

# 计算机应用基础

# L<sup>A</sup>T<sub>E</sub>X 科技排版

潘建瑜

[jypan@math.ecnu.edu.cn](mailto:jypan@math.ecnu.edu.cn)

华东师范大学数学系

2010.03



# 本章内容

- ① T<sub>E</sub>X介绍
- ② L<sup>A</sup>T<sub>E</sub>X介绍
- ③ 中文排版
- ④ 本课程相关信息

# T<sub>E</sub>X是什么？

## □ T<sub>E</sub>X是一个功能强大的排版系统

- ☞ T<sub>E</sub>X系统提供了300多条基本的排版命令
- ☞ 用户可以在这些基本命令的基础上自定义一些新的命令
- ☞ T<sub>E</sub>X系统是公认的**数学公式排得最好的系统**
- ☞ T<sub>E</sub>X是**免费**的，其开发者公开了全部源代码



# $\text{T}_{\text{E}}\text{X}$ 的优点

- ◆ **高质量的输出**:  $\text{T}_{\text{E}}\text{X}$ 以排版质量为首要目标
- ◆ **超常的稳定性**:  $\text{T}_{\text{E}}\text{X}$ 系统极少崩溃
- ◆  **$\text{T}_{\text{E}}\text{X}$ 是宏命令编程语言**: 你可以用很少的命令来完成非常复杂的工作, 或通过重新定义 $\text{T}_{\text{E}}\text{X}$ 命令来得到特殊的效果
- ◆  **$\text{T}_{\text{E}}\text{X}$ 文档是纯文本文件**: 占用空间很小
- ◆ **良好的通用性和低廉的价格**: 免费
- ◆ **超级技术支持**: 丰富的网络资源

# T<sub>E</sub>X的缺点

## ❖ 命令繁多

👉 常备一本参考资料

👉 要多用

## ❖ 错误难找：积累经验

## ❖ 写宏包有难度：普通用户不需要自己写宏包

## ❖ 使用不直观：目前已有一些所见即所得的扩展

# T<sub>E</sub>X的发展历史

- ◆ T<sub>E</sub>X的第一版于1978 年面世  
七十年代末，D. E. Knuth(高德纳)在看到其巨著 “The Art of Computer Programming” 第二卷的校样时，对排版的低质量感到无法忍受，于是决定开发一个高质量的计算机排版系统，这样就有了 T<sub>E</sub>X
- ◆ 1982 年版的 T<sub>E</sub>X 已是一个十分稳定的版本
- ◆ 1993 年左右 Knuth 宣布不再对 T<sub>E</sub>X 进行更新
- ◆ T<sub>E</sub>X 3.1415926 (2008年)  $\rightarrow \pi$



## 常见的 $\text{T}_{\text{E}}\text{X}$ 宏包（宏集、扩展）

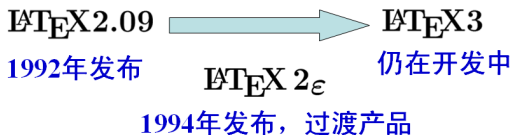
- ❖ **Plain  $\text{T}_{\text{E}}\text{X}$** : Knuth 在  $\text{T}_{\text{E}}\text{X}$  基本命令的基础上，新定义 600 多条复合命令构成，目前使用的  $\text{T}_{\text{E}}\text{X}$  通常是指 Plain  $\text{T}_{\text{E}}\text{X}$
- ❖  **$\mathcal{A}\mathcal{M}\mathcal{S}\text{-}\text{T}_{\text{E}}\text{X}$** : 由美国数学会开发，容易排版复杂的数学公式 (**`amsmath`** 宏包)
- ❖  **$\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$** : 由 L. Lamport 编写，适合排版普通文章和书籍
- ❖  **$\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X} 2_{\epsilon}$** : 可以加载 **`amsmath`** 宏包，目前最流行的  $\text{T}_{\text{E}}\text{X}$  扩展



# L<sup>A</sup>T<sub>E</sub>X发展

❖ L<sup>A</sup>T<sub>E</sub>X 首先由 Lamport 于1985年开发出来，目前由 L<sup>A</sup>T<sub>E</sub>X3 Project 负责开发和维护

❖ L<sup>A</sup>T<sub>E</sub>X 的版本



目前普遍使用的是 L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub>

# L<sup>A</sup>T<sub>E</sub>X 的特点

- ✿ **自动编号**: 章节、图表、公式定理、文献...
- ✿ **自动生成目录、索引**
- ✿ 公式、定理、参考文献、插图、页码等可以进行**交叉引用**
- ✿ 可将输出**直接转化成 pdf 文件**
- ✿ 可建立**参考文献库**
- ✿ 可以通过**添加各种宏包**以扩大其功能, 实现各种特殊要求
- ✿ .....

# L<sup>A</sup>T<sub>E</sub>X 和 Word

- ◆ Word 简单易用，所见即所得
- ◆ 普通办公文档编辑可以使用 Word
- ❖ L<sup>A</sup>T<sub>E</sub>X 稳定，输出美观，质量高
- ❖ 科技论文书籍(特别是数学类)的排版应使用 L<sup>A</sup>T<sub>E</sub>X

L<sup>A</sup>T<sub>E</sub>X 具有 Word 的所有功能，定制性高，  
但易用性不如 Word

# 接受 L<sup>A</sup>T<sub>E</sub>X 投稿的出版机构

- ✿ AMS
- ✿ SIAM
- ✿ Elsevier
- ✿ Springer
- ✿ Cambridge University Press
- ✿ IEEE
- ✿ Kluwer Academic Publishers
- ✿ . . . . .

# $\text{T}_\text{E}\text{X}(\text{L}^\text{A}\text{T}_\text{E}\text{X})$ 的发行版

❑ 发行版：将源代码编译成程序，与其它应用程序一起打包发行

👉 Windows 系统：MikTeX, fpTeX, PCTeX ...

👉 Unix/Linux 系统：teTeX

👉 Mac OS 系统：MacTeX

❑ TeXlive：由国际  $\text{T}_\text{E}\text{X}$  用户组织开发，支持不同的操作系统平台

# 中文排版

## □ 中文预处理系统

☞ **CCT**: 科学院张林波教授开发, 在文档格式方面相当符合中文习惯

☞ **TY**: 由华师大肖刚、陈志杰等教授开发

□ **CJK**: 由德国 W. Lemberg 开发, 可以同时处理中、日、韩三国文字

□ CJK 与 L<sup>A</sup>T<sub>E</sub>X 的结合更为紧密, 但中文化不如 CCT 和 TY

□ 目前中文处理的主流: **L<sup>A</sup>T<sub>E</sub>X + CJK**

# 中文 $\text{T}_{\text{E}}\text{X}$ 套装

- ❑  $\text{C}\text{T}_{\text{E}}\text{X}$ : 基于 MikTeX, 集成 CCT, TY, CJK
- ❑  $\text{C}\text{T}_{\text{E}}\text{Xlive}$ : 基于 Texlive, 支持中文排版
- ❑  $\text{China}\text{T}_{\text{E}}\text{X}$ : 基于 MikTeX, 集成 CCT, TY, CJK
- ☞ 以上套装安装都比较简单 (“傻瓜式” 安装)
- ☞ WinEdt 和 GSView 的注册: 见课程主页

# CT<sub>E</sub>X 套裝

- ◆ 最新稳定版 2.8.0.125
  - CTeX 2.8.0.125.exe, 137M
  - CTeX 2.8.0.125 Full.exe, 756M
- ◆ 2.8.0.125 版包含以下内容
  - **MiKTeX** 2.8
  - **CCT** 0.61803-2a, **CJK**, **TY**
  - **WinEdt** 5.6 beta1
  - **SumatraPDF**, **GSview**, **Ghostscript**
  - **TeXFriend**
- ◆ 一个较老的版本: 2.4.6 版



## 本课程相关信息

- ❑ 教材：《L<sup>A</sup>T<sub>E</sub>X 入门与提高》第二版，2006
- ❑ 课程主页：<http://math.ecnu.edu.cn/~latex>
- ❑ 参考资料：见课程主页
- ❑ 考核方式：平时60%+期末40%
- ❑ 学习建议：

**在使用中学习，不是在记忆中学习**

## 第二章 准备文稿

2010.03



# 主要内容

## ① 编写 T<sub>E</sub>X 源文件: 纯文本, 以 tex 为扩展名

☞ 可以使用任何文本编辑器编写

☞ WinEdt, WinShell, EditPlus, Vi, Emacs

## ② 用 L<sup>A</sup>T<sub>E</sub>X 编译 tex 源文件, 生成 dvi 文件

✿ 如果出现错误, 则返回第一步进行修改

## ③ 输出 dvi 文件: 屏幕或打印机, 或转化为 ps 或 pdf 文件

◆ 由十进制小数加长度单位表示

◆ 常用长度单位

mm	毫米	pt	点
cm	厘米	em	大写字母 M 的宽度
in	英寸	ex	小写字母 x 的高度

☞  $1 \text{ in} = 2.54\text{cm} = 72.27\text{pt}$

☞ em 与 ex 与当前字体尺寸相关

◆ 弹性长度: 根据排版需要自动伸缩

✿ 正常值 plus 伸展值 minus 收缩值

✿ 实际长度可以超过正常值和伸展值之和, 但不能小于正常值和收缩值之差

❖ 文稿 (T<sub>E</sub>X 源文件): 正文 + 排版控制命令

❖ 排版控制命令: 用倒斜线引导的字符串

☞ 控制字: 倒斜线后加一个或多个字母组成, 区分大小写, 如 `\alpha`

☞ 控制符: 倒斜线后加一个特殊字符组成, 如 `\%`

❖ 排版命令中的参数: 可选参数, 不可省略参数

`\排版命令[可选参数]{不可省略的参数}`

☞ 方括号中的参数为可选的

☞ 花括号中的为不可省略的参数

☞ 不带参数的命令有时也称为声明

## ❖ 例: 201.tex

```
\documentclass[11pt]{article}
\begin{document}

Hi, This is my first \LaTeX file.

\end{document}
```

## ◆ 第一行决定文档的版式和基本字体的大小

☞ `article` 表示这是一篇普通文章,  
其它常用的有: `book`, `report`, `letter` 等

☞ `11pt` 表示基本字体大小为11pt,  
也可以改用12pt 或10pt (缺省)

## ◆ `\begin{xxx}` 和 `\end{xxx}` 构成一个环境

## ◆ 一篇文档中有且只能有一个 `document` 环境

## ◆ `\documentclass` 和 `\begin{document}` 之间的区域称为导言区, 用于放置一些全局控制命令



## ❖ 例: 202.tex

```
\documentclass[11pt]{article}
\usepackage{CJK}
\begin{document}
\begin{CJK}{GBK}{song}
```

祝贺你, MikTeX 和 CJK 安装成功了!

```
\end{CJK} \end{document}
```

- ◆ 使用中文时必须在导言区中引入 CJK 宏包

```
\usepackage{CJK}
```

- ◆ 汉字必须放入 CJK 环境中:

```
\begin{CJK}{GBK}{song}
```

☞ GBK 表示所采用的字符集

☞ song 表示默认使用宋体字,  
其它字体还有 kai, fs, hei, li, you

- ◆ CJK 宏包定义了两种环境: CJK 和 CJK\*

```
\begin{CJK*}{GBK}{song}
```

```
\begin{CJK*}{GBK}{kai}
```

# CJK 与 CJK\* 的区别

## ◆ CJK 和 CJK\* 的区别

- ☞ CJK 环境不会自动忽略汉字后面的空格  
CJK 环境中汉字之间不要随意留空
- ☞ CJK\* 环境将自动忽略汉字后面的空格

## ◆ CJK\* 环境中的空格

- ☞ 汉字后面添加空格: \\_ 或使用 ~
- ☞ 其它小间隔命令: \, \; \: \!
- ☞ CJK\* 环境会保留西文后面的空格

## ◆ 多个连续的空格被看成一个空格

◆ 分组:  $\{ \dots \}$

祝贺你,  $\texttt{\textbf{MikTeX 和 CJK}}$  安装成功了!

◆ 注释符:  $\%$

◆ 西文标点后要加空格

◆ 各种环境的开始和结束命令最好独占一行

◆ 换行: 连续两个回车 (单个回车仅为一个空格)

◆ 其它输出:

☞  $\text{dvi} \rightarrow \text{ps}$ ,      $\text{dvi} \rightarrow \text{pdf}$

☞  $\text{ps} \rightarrow \text{pdf}$ ,      $\text{\LaTeX} \rightarrow \text{pdf}$

字符	#	\$	%	{	}	~	_	^	\	&
命令	\#	\\$	\%	\{	\}	\~	\_	\^	\textbackslash	\&

字符		<	>	*	*
命令	\$	\$<	\$>	*	\$* \$

字符	§	¶	†	‡	©	£
命令	\S	\P	\dag	\ddag	\copyright	\pounds

字符	$\mathrm{T}_\mathrm{E}X$	$\mathrm{L}^{\mathrm{A}}\mathrm{T}_\mathrm{E}X$	$\mathrm{L}^{\mathrm{A}}\mathrm{T}_\mathrm{E}X 2_\epsilon$	$\mathcal{A}MS\text{-}\mathrm{L}^{\mathrm{A}}\mathrm{T}_\mathrm{E}X^1$
命令	\TeX	\LaTeX	\LaTeXe	\AmS-\LaTeX

<sup>1</sup>需要调用 `amsmath` 宏包

## □ 引号

- ☞ 左单引号: 用键盘上的倒引号
- ☞ 右单引号: 直接输入键盘上的单引号
- ☞ 左双引号: 连续两个倒引号
- ☞ 右双引号: 直接输入键盘上的双引号

## □ 连字号

- ☞ 输入单个连字号: 产生一个连字符 -
- ☞ 连续两个连字号: 产生表示数字范围的符号 -
- ☞ 连续三个连字号: 产生一个西文破折号 —
- ☞ \$-\$: 数学公式中的减号

## □ 圆点(句号)的两种作用

- 👉 在句尾时表示句子的结束：西文句号
- 👉 表示缩写：圆点

## □ 两种圆点(句号)后面的空格长度不一样

$\text{\TeX}$  自动在句号后插入额外的空白

- 👉 圆点前加 `\@` 表示句号
- 👉 句号后的空格用 `\_` 或 `~`, 则不会加入额外空白
- 👉 通常大写字母后面的圆点看作是缩写

## □ 命令: `\frenchspacing` 取消句号后的空格

## 非英文字母

œ	æ	å	ø	ł	ß	ı	İ
\oe	\ae	\aa	\o	\l	\ss	!\	?\`
Œ	Æ	Å	Ø	Ł	SS	ı	İ
\OE	\AE	\AA	\O	\L	\SS	\i	\j

## 重音符号: 以字母 o 为例

ò	ó	ô	ö	õ	ō	ô	ö
\`o	\'o	\^o	\"o	\~o	\=o	\.o	\u{o}
ö	ő	õ	ôo	ô	ô	ô	
\v{o}	\H{o}	\r{o}	\t{oo}	\b{o}	\c{o}	\d{o}	



- ◆ 分行: L<sup>A</sup>T<sub>E</sub>X 会根据排版需要自动分行
- ◆ 强制分行: `\\` 或 `\\*` 或 `\newline`
  - ☞ `\\` 和 `\\*` 后面可以带长度参数, 以增加当前行与 newRow 之间的距离, 参数可正可负, 如: `\\[3mm]`
- ◆ 建议分行: `\linebreak[n]` (例 204.tex)
  - ☞ `n` 可以取 0, 1, 2, 3, 4, 数越大建议力度越大
  - ☞ `\linebreak` 强制换行, 撑满整行(增加字间距)
  - ☞ `\nolinebreak[n]` 建议不分行
  - ☞ `\mbox{文本内容}` 使文本内容保持在同一行
  - ☞ 可在英文单词中插入 `\-` 建议断词

- 分段: `\par`, 或连续两个回车(空行)
- 分页: 自动分页
  - ◆ 强制分页: `\newpage`
  - ◆ 建议分页: `\pagebreak[n]` , `\nopagebreak[n]`
  - ✿ 增加当前页高度有时可以避免难看的分页
    - 👉 `\enlargethispage{尺寸}` 可增加的最大高度
    - 👉 `\enlargethispage*{尺寸}` 严格指定增加高度

- ◆ `\quad` 插入的空白相当于当前字体尺寸大小
- ◆ `\qqquad`  $\rightarrow$  `\quad` 的两倍
- ◆ `\,`  $\rightarrow$  大约为 `\quad` 的  $3/18$
- ◆ `\hspace{宽度大小}` 和 `\hspace*{宽度大小}`
- ◆ `\hfill`: 弹性长度, 根据排版需要插入空白, 撑满整行 (`\hspace{\hfill}`)
- ◆ `\hphantom{文本内容}`: 占据文本内容的宽度
- ◆ `\vphantom{文本内容}`, `\phantom{文本内容}`

◆ 导引线: `\dotfill`, `\hrulefill`

◆ 垂直间距

☞ `\vspace{高度}` 和 `\vspace*{高度}`

☞ `\vfill`: 相当于 `\vspace{\fill}`

☞ `\smallskip`:  $\rightarrow$  `\vspace{\smallskipamount}`

☞ `\medskip`:  $\rightarrow$  `\vspace{\medskipamount}`

☞ `\bigskip`:  $\rightarrow$  `\vspace{\bigskipamount}`

✿ 上面最后三个长度可以自己重新定义

- ◆ 段落首行缩进: 由 `\parindent` 的值决定, 例

`\setlength{\parindent}{2em}`

- ◆ `\CJKindent`: 首行缩进两个汉字

- ◆ `\indent` 与 `\noindent`

- ♣ 每节的第一段首行不会自动缩进

☞ `\hspace{\parindent}`

☞ 在导言区加入: `\usepackage[indentfirst]`

- ◆ 段落间距: `\lineskip` + `\parskip`

☞ 可用 `\setlength` 命令修改以上距离的值

- ◆ `\baselineskip`: 相邻两行基线之间的距离
- ◆ `\baselinestretch`: 行距伸展因子. 例:

```
\renewcommand{\baselinestretch}{1.2}
```

- ☞ 通常是通过修改伸展因子来改变行距
- ☞ 如果上述命令放在导言区时, 则直接对整篇文章起作用
- ☞ 若放在 `\begin{document}` 之后, 则只有当字体尺寸发生改变时才会起作用, 而且只从字体尺寸命令所在段落开始起作用

- ◆ 中文排版: 在导言区加入 `\usepackage{CJK}`
- ◆ CJK 宏包定义了两种环境: CJK 和 CJK\*

```
\begin{CJK}{编码}{字体族}  
:  
\end{CJK}
```

```
\begin{CJK*}{编码}{字体族}  
:  
\end{CJK*}
```

- 👉 编码: GB, GBK, Bg5, Gbt
- 👉 字体族: song, fs, kai, hei, li, you

- ◆ CJK 和 CJK\* 环境之间的切换:

\CJKspace 和 \CJKnospace

- ◆ 在正文中更换字体: \CJKfamily{字体族}

- ♣ 为了使用方便, 通常自定义以下命令:

```
\newcommand{\song}{\CJKfamily{song}}  
\newcommand{\fs}{\CJKfamily{fs}}  
\newcommand{\hei}{\CJKfamily{hei}}  
\newcommand{\kai}{\CJKfamily{kai}}  
\newcommand{\li}{\CJKfamily{li}}  
\newcommand{\you}{\CJKfamily{you}}
```

(例: 205.tex)



# ◆ \CJKtilde: 重新定义 ~ 的长度

```
\def~{\hspace{0.25em plus 0.125em minus 0.07em}}
```

☞ 当西文或数学公式夹在汉字中间时, 在交界处插入波浪号

☞ \standardtilde: 恢复波浪号的原来定义

☞ \nbs: 原波浪号的长度

# ◆ \CJKglue: 必要时在汉字之间插入一个小间隔, 以解决行的超长问题

# ◆ \CJKkern: 禁止在所在的两汉字间换行

◆ 如果编译出错, 会在屏幕上显示一个问号: ?  
并暂停编译, 等待用户输入命令

☞ 回车: 继续处理

☞ X: 退出

☞ ?: 显示可以使用的命令

☞ S: 忽略后续错误

☞ R: 执行下去, 出现错误也不停留

☞ Q: 不提示任何消息地执行

☞ I: 插入一些内容

☞ E: 编辑源文件

☞ H: 显示帮助

◆ 更多 CJK 的用法请参见:

`$CTeX\texmf\doc\latex\CJK\`

目录下的 `commmands.doc` 和 `CJK.doc` 文档

# 第三章 文字模式

潘建瑜

华东师范大学数学系

2010.03



# 主要内容

- 1 西文字体
- 2 中文字体
- 3 参考文献
- 4 表格
- 5 脚注

# L<sup>A</sup>T<sub>E</sub>X 排版的三种模式

□ 段落模式: 自动分行, 分段, 分页.

# L<sup>A</sup>T<sub>E</sub>X 排版的三种模式

□ 段落模式: 自动分行, 分段, 分页.

□ 左到右模式:

将输入字符排成一行, 不换行, 无论长短

# L<sup>A</sup>T<sub>E</sub>X 排版的三种模式

□ 段落模式: 自动分行, 分段, 分页.

□ 左到右模式:

将输入字符排成一排, 不换行, 无论长短

♣ 以上两种模式统称为 文字模式



# L<sup>A</sup>T<sub>E</sub>X 排版的三种模式

□ 段落模式: 自动分行, 分段, 分页.

□ 左到右模式:

将输入字符排成一排, 不换行, 无论长短

♣ 以上两种模式统称为 文字模式

□ 数学模式: 排版数学公式

- 1 西文字体
- 2 中文字体
- 3 参考文献
- 4 表格
- 5 脚注

# 西文字体属性

□ 字体有 5 种属性: 编码, 族, 系列, 形状, 大小

# 西文字体属性

□ 字体有 5 种属性: 编码, 族, 系列, 形状, 大小

◆ 族 (family): 字体的概观样式

☞ `\rmfamily`: 罗马 (roman) 字体

☞ `\sffamily`: 无衬线 (sans serif) 字体

☞ `\ttfamily`: 打字机 (typewriter) 字体

# 西文字体属性

❑ 字体有 5 种属性: 编码, 族, 系列, 形状, 大小

◆ 族 (family): 字体的概观样式

☞ `\rmfamily`: 罗马 (roman) 字体

☞ `\sffamily`: 无衬线 (sans serif) 字体

☞ `\ttfamily`: 打字机 (typewriter) 字体

◆ 形状 (shape): 字体的高矮和倾斜

☞ `\upshape`: 直立 (upstanding) 字体

☞ `\itshape`: 意大利斜体 (*italic*)

☞ `\slshape`: (*slanted*) 斜体

☞ `\scshape`: 小体大写 (SMALL CAPS) 字体

# 西文字体属性

◆ 系列 (series): 字体的宽度和权重 (黑度)

☞ `\mdseries`: 中等权重

☞ `\bfseries`: 粗体 (Bold face)

# 西文字体属性

## ◆ 系列 (series): 字体的宽度和权重 (黑度)

☞ `\mdseries`: 中等权重

☞ `\bfseries`: 粗体 (Bold face)

## ♣ 以上关于字体的命令都称为**声明**

☞ 这些命令在遇到新的同一类命令之前一直起作用.

☞ 若想限制声明的作用范围, 可使用**组**或**相应环境**

# 西文字体属性

◆ 系列 (series): 字体的宽度和权重 (黑度)

☞ `\mdseries`: 中等权重

☞ `\bfseries`: 粗体 (Bold face)

♣ 以上关于字体的命令都称为**声明**

☞ 这些命令在遇到新的同一类命令之前一直起作用.

☞ 若想限制声明的作用范围, 可使用**组**或**相应环境**

◆ `\normalfont`:

☞ 缺省字体, 即中等权重, 直立, 罗马字体



# 西文字体属性

◆ 系列 (series): 字体的宽度和权重 (黑度)

☞ `\mdseries`: 中等权重

☞ `\bfseries`: 粗体 (Bold face)

♣ 以上关于字体的命令都称为**声明**

☞ 这些命令在遇到新的同一类命令之前一直起作用.

☞ 若想限制声明的作用范围, 可使用**组**或**相应环境**

◆ `\normalfont`:

☞ 缺省字体, 即中等权重, 直立, 罗马字体

◆ 西文字体基本尺寸有: 10pt, 11pt 和 12pt

# 字体命令

◆ 每一个**字体声明**都有对应的**字体命令**

# 字体命令

- ◆ 每一个**字体声明**都有对应的**字体命令**
- ◆ 字体命令: 只对其参数中的文本起作用, 如:

```
\textbf{这是黑体}
```

# 字体命令

- ◆ 每一个**字体声明**都有对应的**字体命令**
- ◆ 字体命令: 只对其参数中的文本起作用, 如:

```
\textbf{这是黑体}
```

- 👉 族: `\textrm{...}`, `\textsf{...}`, `\texttt{...}`
- 👉 形状: `\textup{...}`, `\textit{...}`,  
`\textsl{...}`, `\textsc{...}`
- 👉 系列: `\textmd{...}`, `\textbf{...}`
- 👉 默认值: `\textnormal{...}`
- 👉 强调: `\emph{...}`

# 字体命令

◆ 每一个**字体声明**都有对应的**字体命令**

◆ 字体命令: 只对其参数中的文本起作用, 如:

```
\textbf{这是黑体}
```

👉 族: `\textrm{...}`, `\textsf{...}`, `\texttt{...}`

👉 形状: `\textup{...}`, `\textit{...}`,  
`\textsl{...}`, `\textsc{...}`

👉 系列: `\textmd{...}`, `\textbf{...}`

👉 默认值: `\textnormal{...}`

👉 强调: `\emph{...}`

◆ 强调命令所对应的声明为 `\em`

# 选择字体尺寸

- ◆ 定义了字体的基本尺寸后, 可使用下面的声明来改变字体大小

<code>\tiny</code>	5pt	<code>\large</code>	12pt
<code>\scriptsize</code>	7pt	<code>\Large</code>	14.4pt
<code>\footnotesize</code>	8pt	<code>\LARGE</code>	17.28pt
<code>\small</code>	9pt	<code>\huge</code>	20.74pt
<code>\normalsize</code>	10pt	<code>\Huge</code>	24.88pt

# 选择字体尺寸

- ◆ 定义了字体的基本尺寸后, 可使用下面的声明来改变字体大小

<code>\tiny</code>	5pt	<code>\large</code>	12pt
<code>\scriptsize</code>	7pt	<code>\Large</code>	14.4pt
<code>\footnotesize</code>	8pt	<code>\LARGE</code>	17.28pt
<code>\small</code>	9pt	<code>\huge</code>	20.74pt
<code>\normalsize</code>	10pt	<code>\Huge</code>	24.88pt

注: 以上尺寸是基本尺寸为 10pt 时的大小  
当基本尺寸为其它尺寸时, 各字体大小见 301.tex

- 1 西文字体
- 2 中文字体
- 3 参考文献
- 4 表格
- 5 脚注



# 中文字体

□ 所有的西文字体命令对中文同样起作用

# 中文字体

□ 所有的西文字体命令对中文同样起作用

👉 *italic* 和 *slanted* 斜体对中文的效果一样

# 中文字体

□ 所有的西文字体命令对中文同样起作用

☞ *italic* 和 *slanted* 斜体对中文的效果一样

□ 中文书籍的基本字号为五号 (10.5磅  $\approx$  11pt)

# 中文字体

□ 所有的西文字体命令对中文同样起作用

☞ *italic* 和 *slanted* 斜体对中文的效果一样

□ 中文书籍的基本字号为五号 (10.5磅  $\approx$  11pt)

□ 最大字体尺寸 \Huge 略小于一号汉字

# 中文字体

❑ 所有的西文字体命令对中文同样起作用

☞ *italic* 和 *slanted* 斜体对中文的效果一样

❑ 中文书籍的基本字号为五号 (10.5磅  $\approx$  11pt)

❑ 最大字体尺寸 \Huge 略小于一号汉字

❑ 使用 GBK 编码时, 可同时输入简体字和繁体字, 各种冷僻字只要能输入就能排版出.  
如: 華東師範大學, 硃鎔基

# 中文字体

❑ 所有的西文字体命令对中文同样起作用

☞ *italic* 和 *slanted* 斜体对中文的效果一样

❑ 中文书籍的基本字号为五号 (10.5磅  $\approx$  11pt)

❑ 最大字体尺寸 \Huge 略小于一号汉字

❑ 使用 GBK 编码时, 可同时输入简体字和繁体字, 各种冷僻字只要能输入就能排版出.

如: 華東師範大學, 硃鎔基

❑ CJKspace 宏包: 忽略汉字之间的空格, 保留汉字与英文 English 之间的空格

# 中文字体

## □ 自定义字体大小: 字体选择命令

```
\fontsize{字体尺寸}{行距}
```

# 中文字体

## ❑ 自定义字体大小: 字体选择命令

```
\fontsize{字体尺寸}{行距}
```

- ☞ 字体选择命令后面必须加上 `\selectfont` 后才生效
- ☞ 行距通常为字体大小的 1.2 – 1.5倍
- ☞ 可得到任意大小的汉字, 但西文最大不能超过 `\Huge`

```
\fontsize{42pt}{\baselineskip}\selectfont
```



# 中文字体

## ❑ 自定义字体大小: 字体选择命令

```
\fontsize{字体尺寸}{行距}
```

- ☞ 字体选择命令后面必须加上 `\selectfont` 后才生效
- ☞ 行距通常为字体大小的 1.2 – 1.5倍
- ☞ 可得到任意大小的汉字, 但西文最大不能超过 `\Huge`

```
\fontsize{42pt}{\baselineskip}\selectfont
```

- ## ❑ `\usepackage{type1cm}` (或 `a0size` 宏包):
- 可得到任意大小的西文字体 (例: 301.tex)

# 中文字号

初号 = 42磅 = 14.82毫米

小初 = 36磅 = 12.70毫米

一号 = 26磅 = 9.17毫米

小一 = 24磅 = 8.47毫米

二号 = 22磅 = 7.76毫米

小二 = 18磅 = 6.35毫米

三号 = 16磅 = 5.64毫米

小三 = 15磅 = 5.29毫米

1英寸 = 72磅 = 2.54cm

四号 = 14磅 = 4.94毫米

小四 = 12磅 = 4.23毫米

五号 = 10.5磅 = 3.70毫米

小五 = 9磅 = 3.18毫米

六号 = 7.5磅 = 2.56毫米

小六 = 6.5磅 = 2.29毫米

七号 = 5.5磅 = 1.94毫米

八号 = 5磅 = 1.76毫米

# 文本居中

- ◆ 文本默认格式是左对齐


# 文本居中

- ◆ 文本默认格式是左对齐
- ◆ 单行文本居中: `\centerline{...}`

# 文本居中

- ◆ 文本默认格式是左对齐
- ◆ 单行文本居中: `\centerline{...}`
- ◆ 多行文本居中: 使用 `center` 环境

```
\begin{center}  
:  
\end{center}
```

 居中环境中每行文本都居中显示

# 文本居中

- ◆ 文本默认格式是左对齐
- ◆ 单行文本居中: `\centerline{...}`
- ◆ 多行文本居中: 使用 `center` 环境

```
\begin{center}  
:  
\end{center}
```

 居中环境中每行文本都居中显示

- ◆ 居中声明: `\centering` (建议少用)

- 1 西文字体
- 2 中文字体
- 3 参考文献
- 4 表格
- 5 脚注

# 参考文献

## □ 参考文献环境: `thebibliography`

```
\begin{thebibliography}{编号样本}  
\bibitem[记号]{引用标志} 文献条目  
:  
\bibitem[记号]{引用标志} 文献条目  
\end{thebibliography}
```



# 参考文献

## □ 参考文献环境: `thebibliography`

```
\begin{thebibliography}{编号样本}
\bibitem[记号]{引用标志} 文献条目
:
\bibitem[记号]{引用标志} 文献条目
\end{thebibliography}
```

[1] 陈志杰等,  $\text{\LaTeX}$  入门与提高, ...

[2] 郭力, 张林波, 葛向阳, CCT中外文科技...

[JS Deng] 邓建松,  $\text{\LaTeX}$  2 $\epsilon$  科技排版指南...

(例 303.tex)

# 参考文献的说明

`\bibitem`[记号]{引用标志} 文献条目

## ◆ “记号” 是可选选项

- ☞ 若没有“记号”选项, 则自动生成带方括号的编号; 当文献增加或减少时, 编号会自动改变

# 参考文献的说明

`\bibitem`[记号]{引用标志} 文献条目

## ◆ “记号” 是可选选项

- ☞ 若没有“记号”选项, 则自动生成带方括号的编号; 当文献增加或减少时, 编号会自动改变
- ☞ 若使用“记号”, 则该文献不被自动编号, 在编号位置和正文中引用该文献的地方显示该记号

# 参考文献的说明

`\bibitem`[记号]{引用标志} 文献条目

## ◆ “记号” 是可选选项

- ☞ 若没有“记号”选项, 则自动生成带方括号的编号; 当文献增加或减少时, 编号会自动改变
- ☞ 若使用“记号”, 则该文献不被自动编号, 在编号位置和正文中引用该文献的地方显示该记号
- ☞ “编号样本”用于指定记号的宽度, 通常为“记号”和文献编号的最大宽度
- ☞ “编号样本”一般为数字, 其位数为最大编号的位数

# 参考文献的说明

`\bibitem`[记号]{引用标志} 文献条目

## ◆ 引用标志不可省略

 引用标志可以由字母, 数字和除逗号外的符号组成

# 参考文献的说明

`\bibitem`[记号]{引用标志} 文献条目

## ◆ 引用标志不可省略

- ☞ 引用标志可以由字母, 数字和除逗号外的符号组成
- ☞ 各个文献的引用标志互不相同

# 参考文献的说明

`\bibitem`[记号]{引用标志} 文献条目

## ◆ 引用标志不可省略

- 👉 引用标志可以由字母, 数字和除逗号外的符号组成
- 👉 各个文献的引用标志互不相同
- 👉 引用文献时使用 `\cite` 命令, 如:

`\cite{引用标志}`, `\cite{引用标志1, 引用标志2}`

`\cite{引用标志1, 引用标志2, ...}`

# 文献间距和文献标题

`\bibitem`[记号]{引用标志} 文献条目

## ◆ 调节各文献之间的距离:

`\setlength{\itemsep}`{高度}



# 文献间距和文献标题

`\bibitem`[记号]{引用标志} 文献条目

## ◆ 调节各文献之间的距离:

`\setlength{\itemsep}`{高度}

## ◆ 参考文献的标题由 `\refname` 的值来确定

☞ 缺省为靠左对齐的“Reference”

`(\Large\bfseries)`

☞ 可以通过修改 `\refname` 的值来更改参考文献标题

☞ 书籍的参考文献标题由“`\bibname`”确定

# 文献条目

`\bibitem`[记号]{引用标志} 文献条目

# 文献条目

`\bibitem`[记号]{引用标志} 文献条目

## ◆ 文献条目通常包含:

作者, 标题, 出版社, 年代, 版本, 页码等

☞ 一行放不下时, 后面的行会自动缩进, 缩进距离为“**编号样本**”的宽度

# 文献条目

`\bibitem`[记号]{引用标志} 文献条目

## ◆ 文献条目通常包含:

作者, 标题, 出版社, 年代, 版本, 页码等

☞ 一行放不下时, 后面的行会自动缩进, 缩进距离为“**编号样本**”的宽度

## ◆ 如果文档中带参考文献, 编译时需**连续编译两遍**

- 1 西文字体
- 2 中文字体
- 3 参考文献
- 4 表格**
- 5 脚注

# 表格

## ◆ 表格环境: `tabular`

```
\begin{tabular}[竖向位置]{列格式}
```

第一行

⋮

第末行

```
\end{tabular}
```

# 表格

## ◆ 表格环境: `tabular`

```
\begin{tabular}[竖向位置]{列格式}
```

第一行

⋮

第末行

```
\end{tabular}
```

- 👉 **竖向位置**: 表格在竖直方向与外部文本行的相对位置, 取值可以是: `t` 或 `b`, 缺省为**居中**
- 👉 表格每行由若干列组成, 列与列之间用 `&` 分隔, 每行必须用 `\\` 结束. (例 304.tex)

# 表格

```
\begin{tabular}{||c|c|c||}
```

◆ **列格式**: 由若干项组成, 用于指定各列的格式

☞ 指定文本位置: **l**, **c**, **r** → 左对齐, 居中, 右对齐

☞ 边界线: **|**, **||**, ...



# 表格

```
\begin{tabular}{||c|c|c||}
```

◆ **列格式**: 由若干项组成, 用于指定各列的格式

☞ 指定文本位置: `l`, `c`, `r` → 左对齐, 居中, 右对齐

☞ 边界线: `|`, `||`, `\cdots`

◆ **画横线**:

☞ `\hline`: 与表格同宽的水平线

☞ `\cline{m-n}`: 从第 `m` 列开始到第 `n` 列结束的水平线

# 表格

```
\begin{tabular}{||c|c|c||}
```

◆ **列格式**: 由若干项组成, 用于指定各列的格式

☞ 指定文本位置: `l`, `c`, `r` → 左对齐, 居中, 右对齐

☞ 边界线: `|`, `||`, `\cdots`

◆ **画横线**:

☞ `\hline`: 与表格同宽的水平线

☞ `\cline{m-n}`: 从第 `m` 列开始到第 `n` 列结束的水平线

◆ **画竖线**: `\vline` → 在当前位置画一与行等高的竖线

# 合并相邻几列

## ◆ 合并相邻多列:

`\multicolumn{列数}{列格式}{文本内容}`

# 合并相邻几列

## ◆ 合并相邻多列:

`\multicolumn{列数}{列格式}{文本内容}`

☞ 当列数 = 1 时, 可以改变当前列的对齐方式

☞ 例 304.tex

# 合并相邻几列

## ◆ 合并相邻多列:

`\multicolumn{列数}{列格式}{文本内容}`

☞ 当列数 = 1 时, 可以改变当前列的对齐方式

☞ 例 304.tex

## ◆ 改变各行之间的间隔: `\\[长度]`

# 合并相邻几列

## ◆ 合并相邻多列:

`\multicolumn{列数}{列格式}{文本内容}`

☞ 当列数 = 1 时, 可以改变当前列的对齐方式

☞ 例 304.tex

## ◆ 改变各行之间的间隔: `\\[长度]`

## ◆ 表格高级功能及使用技巧见教材第 115 页

- 1 西文字体
- 2 中文字体
- 3 参考文献
- 4 表格
- 5 脚注**

# 脚注

◆ 脚注命令: `\footnote{脚注内容}`



# 脚注

◆ 脚注命令: `\footnote{脚注内容}`

- ☞ 该命令应紧接在需要注释的文字后面, 排版后会在所在处显示一个脚注标记, 并将脚注内容显示在当前页的底部, 并带有相同的脚注标记
- ☞ 脚注标记是上标形式的数字, 并自动编号

# 脚注

## ◆ 脚注命令: `\footnote{脚注内容}`

- 🔊 该命令应紧接在需要注释的文字后面, 排版后会在所在处显示一个脚注标记, 并将脚注内容显示在当前页的底部, 并带有相同的脚注标记
- 🔊 脚注标记是上标形式的数字, 并自动编号
- 🔊 在 article 文档类中, 整篇文章的脚注同一编号
- 🔊 在 book 和 report 文档类中, 每章的脚注统一编号

# 脚注

## ◆ 脚注命令: `\footnote{脚注内容}`

- ☞ 该命令应紧接在需要注释的文字后面, 排版后会在所在处显示一个脚注标记, 并将脚注内容显示在当前页的底部, 并带有相同的脚注标记
- ☞ 脚注标记是上标形式的数字, 并自动编号
- ☞ 在 article 文档类中, 整篇文章的脚注同一编号
- ☞ 在 book 和 report 文档类中, 每章的脚注统一编号
- ☞ 脚注只能位于普通文本中, 不能位于数学模式, 表格, LR盒子等中

# 脚注

## ◆ 脚注命令: `\footnote{脚注内容}`

- 🔔 该命令应紧接在需要注释的文字后面, 排版后会在所在处显示一个脚注标记, 并将脚注内容显示在当前页的底部, 并带有相同的脚注标记
- 🔔 脚注标记是上标形式的数字, 并自动编号
- 🔔 在 article 文档类中, 整篇文章的脚注同一编号
- 🔔 在 book 和 report 文档类中, 每章的脚注统一编号
- 🔔 脚注只能位于普通文本中, 不能位于数学模式, 表格, LR盒子等中
- 🔔 例: 305.tex

# 第四章 数学公式

潘建瑜

华东师范大学数学系

2010.03



## 1 概述

## 2 行内公式与行间公式

## 3 常见数学公式排版

## 4 可以变大的定界符

## 5 数学字体和数学符号表

- ❑ 在导言区加入数学公式宏包：  
`latexsym, bm, amsmath, amssymb`

```
\usepackage{latexsym}  
\usepackage{bm}  
\usepackage{amsmath}  
\usepackage{amssymb}
```

或

```
\usepackage{latexsym,bm,amsmath,amssymb}
```

或

```
\usepackage{latexsym,bm}  
\usepackage{amsmath,amssymb}
```

# 几点说明

数学公式

潘建瑜

概述

行内公式与行间  
公式

常见数学公式排  
版

可以变大的定界  
符

数学字体和数学  
符号表

- 👉 数学模式中的变量为 *italic* 斜体字母
- 👉 数学模式中的空格全部被忽略,  $\mathrm{T_E X}$  自动安排公式各部分间距
- 👉 行内公式、行间公式 (显示公式)
- 👉 数学字体的大小: 自动选取, 也可强制规定

`\displaystyle` → 显示样式

`\textstyle` → 正文样式

`\scriptstyle` → 角标样式

`\scriptscriptstyle` → 二级角标样式



# 几点说明

数学公式

潘建瑜

概述

行内公式与行间公式

常见数学公式排版

可以变大的定界符

数学字体和数学符号表

☞ 可以直接输入的数学运算符号有：

$+$   $-$   $*$   $/$   $=$   $<$   $>$   $( )$   $[ ]$   $|$   $'$   $!$   $:$

☞ 其它符号的输入：

{	}	...	...	⋮	⋱	普通冒号
\{	\}	\ldots	\cdots	\vdots	\ddots	\colon

☞ 数学模式中的普通文本 (英, 汉) 需放在盒子里

$\text{\mbox{普通文本}}$

# 几点说明

数学公式

潘建瑜

概述

行内公式与行间  
公式

常见数学公式排  
版

可以变大的定界  
符

数学字体和数学  
符号表

## □ 数学模式中的空白间隔

☞  $\backslash quad, \backslash qquad$

☞  $\backslash hspace$

☞  $\backslash phantom$

☞  $\backslash$

☞  $\backslash,$  (大小约为  $3/18 \backslash quad$ )

☞  $\backslash;$  →  $5/18 \backslash quad$

$\backslash:$  →  $4/18 \backslash quad$

$\backslash!$  →  $-3/18 \backslash quad$

→ 只能用于数学模式

数学公式

潘建瑜

概述

行内公式与行间  
公式

常见数学公式排  
版

可以变大的定界  
符

数学字体和数学  
符号表

## 1 概述

## 2 行内公式与行间公式

## 3 常见数学公式排版

## 4 可以变大的定界符

## 5 数学字体和数学符号表

# 行内公式与行间公式

数学公式

潘建瑜

概述

行内公式与行间公式

常见数学公式排版

可以变大的定界符

数学字体和数学符号表

□ 行内公式：与普通文本出现在同一行中

☞ `\begin{math} \dots \end{math}`

☞ `\( \dots \)`

☞ `$ \dots $`

勾股定理  $a^2 + b^2 = c^2$  也称商高定理

(推荐使用最后一种方式)

# 行内公式与行间公式

数学公式

潘建瑜

概述

行内公式与行间  
公式

常见数学公式排  
版

可以变大的定界  
符

数学字体和数学  
符号表

## □ 行间公式

- 👉 独占一行(单行公式)或多行(多行公式)
- 👉 行间公式可以编号,也可以不编号  
编号可以人工编号,也可以自动编号
- 👉 行间公式中不能出现空行

# 单行公式

数学公式

潘建瑜

概述

行内公式与行间公式

常见数学公式排版

可以变大的定界符

数学字体和数学符号表

```
\begin{displaymath}
```

... ..

```
\end{displaymath}
```

不带编号

```
\[ ... \]
```

上面的简化形式

```
\begin{equation}
```

... ..

```
\end{equation}
```

自动编号

```
$$ ... $$
```

不自动编号  
可用 `\eqno` 或  
`\leqno` 人工编号

# 公式自动编号 equation 环境

数学公式

潘建瑜

概述

行内公式与行间公式

常见数学公式排版

可以变大的定界符

数学字体和数学符号表

- ☞ **equation** 环境中的公式可以是普通的单行公式, 也可以是作为一个整体处理的环境或盒子
- ☞ 若要改变公式自动编号的值, 可在公式前插入

`\setcounter{equation}{整数}`

这里的 equation 为公式**计数器**, 其后面的公式的编号自动为计数器的值加 1

- ☞ **标记**: `\label{公式标记}` → 公式标志唯一
- ☞ **引用**: `\ref{公式标记}` 和 `\eqref{公式标记}`

# 多行公式

数学公式

潘建瑜

概述

行内公式与行间公式

常见数学公式排版

可以变大的定界符

数学字体和数学符号表

```
\begin{eqnarray}
```

... ..

```
\end{eqnarray}
```

自动编号

```
\begin{eqnarray*}
```

... ..

```
\end{eqnarray*}
```

不带编号

- 紧跟在行间公式后面的文本不会自动缩进
- 每行公式以 `\\` 结束 (最后一行除外)
- 修改单个行间距: `\\[高度]`



# 公式自动编号 `eqnarray` 环境

数学公式

潘建瑜

概述

行内公式与行间公式

常见数学公式排版

可以变大的定界符

数学字体和数学符号表

👉 `eqnarray` 环境中每行公式都自动编号

👉 若某行公式无需编号, 可在该行公式后面的换行符前查入 `\nonumber`

👉 `eqnarray` 环境中每行的格式为

左边公式 & 中间公式 & 右边公式
--------------------

👉 总是按三列排版公式, 每行至多三列

👉 三列的对齐方式分别为: 左对齐, 居中, 右对齐

👉 同一行中各列之间用 `&` 隔开

👉 `eqnarray` 环境内部修改计数器 `equation` 的值

数学公式

潘建瑜

概述

行内公式与行间公式

常见数学公式排版

可以变大的定界符

数学字体和数学符号表

## 1 概述

## 2 行内公式与行间公式

## 3 常见数学公式排版

## 4 可以变大的定界符

## 5 数学字体和数学符号表

# 角标: 上标和下标

数学公式

潘建瑜

概述

行内公式与行间公式

常见数学公式排版

可以变大的定界符

数学字体和数学符号表

□ 上标:  $\wedge\{\cdots\}$ ; 下标:  $\_ \{\cdots\}$

🔔 这两个角标命令必须在**数学模式**中使用

🔔 多层角标需要使用分组符号

🔔 一级角标字体大小约 7pt,  
二级及以上角标字体大小约为 5pt

🔔 中文角标要放入盒子, 并指定字体大小

$$\$ x^{\{\backslash\mathrm{mbox}\{\backslashscriptsize \text{平方}\}\}} \$$$

🔔 可以根据需要手工改变角标字体大小或层次

🔔 **导数**符号:  $\backslashprime$  或 右单引号

# 分式与根式

数学公式

潘建瑜

概述

行内公式与行间公式

常见数学公式排版

可以变大的定界符

数学字体和数学符号表

□ 分式: `/` 或 `\frac{分子}{分母}`

- ☞ 自动选取分子分母字体大小, 也可人工指定
- ☞ 可在分子或分母两边加空格来延长分数线

□ 根式: `\sqrt{···}`, `\sqrt[n]{···}`

- ☞ 不带横线的根号命令: `\surd`
- ☞ 被开方式字符高度不同时, 根号线会在不同水平线上
- ☞ 一些微调技术见教材

# 求和与积分

数学公式

潘建瑜

概述

行内公式与行间公式

常见数学公式排版

可以变大的定界符

数学字体和数学符号表

◆ 求和  $\rightarrow$  `\sum`      积分  $\rightarrow$  `\int`

👉 带上下限: 位于符号右侧, 或符号头顶和脚下

👉 在行内公式和行间公式中取不同的尺寸

👉 可以人工指定上下限的位置:

`\limits`      `\nolimits`

♣ 积分表达式与微分算子之间要留一小空

♣ 微分算子 `d` 为直立字体  $\leftarrow$  `\mathrm{d}`

♣ `\oint`  $\rightarrow$   $\oint$

# 直立的积分号

数学公式

潘建瑜

概述

行内公式与行间  
公式

常见数学公式排  
版

可以变大的定界  
符

数学字体和数学  
符号表

## ◆ 直立的积分号

 `\usepackage{wasysym}`

(需在 `amsmath` 之后调用)

`\varint`  $\rightarrow \int$       例:  $\int_a^b f(x) dx$

`\iint`  $\rightarrow \iint$

`\iiint`  $\rightarrow \iiint$

`\varoint`  $\rightarrow \oint$

`\oiint`  $\rightarrow \oiint$

# 数学重音符号

数学公式

潘建瑜

概述

行内公式与行间公式

常见数学公式排版

可以变大的定界符

数学字体和数学符号表

`\hat{a}`  $\rightarrow \hat{a}$

`\dot{a}`  $\rightarrow \dot{a}$

`\tilde{a}`  $\rightarrow \tilde{a}$

`\acute{a}`  $\rightarrow \acute{a}$

`\grave{a}`  $\rightarrow \grave{a}$

`\bar{a}`  $\rightarrow \bar{a}$

`\ddot{a}`  $\rightarrow \ddot{a}$

`\vec{a}`  $\rightarrow \vec{a}$

`\check{a}`  $\rightarrow \check{a}$

`\mathring{a}`  $\rightarrow \mathring{a}$

`\imath`  $\rightarrow \imath$

`\jmath`  $\rightarrow \jmath$

`\widehat{abc}`  $\rightarrow \widehat{abc}$

`\widetilde{abc}`  $\rightarrow \widetilde{abc}$

# 上、下划线等

数学公式

潘建瑜

概述

行内公式与行间  
公式

常见数学公式排  
版

可以变大的定界  
符

数学字体和数学  
符号表

◆ `\overline`, `\underline`

$$\overline{a^2 + \underline{ab} + \bar{b}^2}$$

◆ `\overbrace`, `\underbrace`

$$\underbrace{a + \overbrace{b + \cdots + b}^{m\uparrow} + c}_{20\uparrow}$$



# 堆积符号

数学公式

潘建瑜

概述

行内公式与行间  
公式

常见数学公式排  
版

可以变大的定界  
符

数学字体和数学  
符号表

$$\vec{x} \stackrel{\text{def}}{=} (x_1, \dots, x_n)$$

◆ `\stackrel`{上位符号}{基位符号}

☞ 基位符号为正常字体，上位符号为较小字体

◆ {上位符号`\atop` 下位符号}

{上位符号`\choose` 下位符号}

☞ 上位与下位符号采用同样大小的字体

$$\sum_{\substack{1 \leq i \leq n \\ i \neq j}} \rightarrow \sum$$

$$\{n+1 \choose k\} \rightarrow \binom{n+1}{k}$$

数学公式

潘建瑜

概述

行内公式与行间公式

常见数学公式排版

可以变大的定界符

数学字体和数学符号表

## 1 概述

## 2 行内公式与行间公式

## 3 常见数学公式排版

## 4 可以变大的定界符

## 5 数学字体和数学符号表

# 可以变大的定界符

数学公式

潘建瑜

概述

行内公式与行间公式

常见数学公式排版

可以变大的定界符

数学字体和数学符号表

$( ) [ ] | / \{ \} || \lfloor \rfloor \lceil \rceil \langle \rangle \uparrow \downarrow \updownarrow \Uparrow \Downarrow \Updownarrow$

◆ `\big` → 放大 1.5 倍. 例: `\big(` →  $($

◆ `\Big` → 放大 2 倍. 例: `\Big(` →  $($

◆ `\bigg` → 放大 2.5 倍. 例: `\bigg(` →  $($

◆ `\Bigg` → 放大 3 倍. 例: `\Bigg(` →  $($

# 可以变大的定界符

数学公式

潘建瑜

概述

行内公式与行间公式

常见数学公式排版

可以变大的定界符

数学字体和数学符号表

---

$$( ) [ ] | / \{ \} \parallel [ ] [ ] [ ] \langle \rangle \uparrow \downarrow \updownarrow \Uparrow \Downarrow \Updownarrow$$

---

---

`\big` → 1.5 倍

$$( ) [ ] | / \{ \} \parallel [ ] [ ] [ ] \langle \rangle \uparrow \downarrow \updownarrow \Uparrow \Downarrow \Updownarrow$$

`\Big` → 2 倍

$$( ) [ ] | / \{ \} \parallel [ ] [ ] [ ] \langle \rangle \uparrow \downarrow \updownarrow \Uparrow \Downarrow \Updownarrow$$

`\bigg` → 2.5 倍

$$( ) [ ] | / \{ \} \parallel [ ] [ ] [ ] \langle \rangle \uparrow \downarrow \updownarrow \Uparrow \Downarrow \Updownarrow$$

`\Bigg` → 3 倍

$$( ) [ ] | / \{ \} \parallel [ ] [ ] [ ] \langle \rangle \uparrow \downarrow \updownarrow \Uparrow \Downarrow \Updownarrow$$

# 可以变大的定界符

数学公式

潘建瑜

概述

行内公式与行间公式

常见数学公式排版

可以变大的定界符

数学字体和数学符号表

## □ 自适应放大命令: `\left`, `\right`

☞ 根据定界符所包含的内容自动放大

☞ 必须成对出现

☞ 必须出现在公式的同一行

☞ `\left.` 和 `\right.`

$$\left(\sum_{i=1}^n i^2\right) \rightarrow \left(\sum_{i=1}^n i^2\right)$$

$$\left.\frac{\partial f}{\partial x}\right|_{x=0}$$

$$\rightarrow \frac{\partial f}{\partial x}\bigg|_{x=0}$$

# 阵列

数学公式

潘建瑜

概述

行内公式与行间  
公式

常见数学公式排  
版

可以变大的定界  
符

数学字体和数学  
符号表

## □ 阵列环境: `array`

```
\begin{array}[竖向位置][列格式]  
第一行 \\  
... \\  
第 n 行 \\  
\end{array}
```

- 🔊 **竖向位置** 和 **列格式** 的含义同 `tabular` 环境
- 🔊 每行都以强制换行符结束, 最后一行除外
- 🔊 每行由多列组成, 各行中的列数相同
- 🔊 该环境只能用于数学模式中

# array 与 eqnarray 的区别

数学公式

潘建瑜

概述

行内公式与行间公式

常见数学公式排版

可以变大的定界符

数学字体和数学符号表

- ☞ `eqnarray` 本身就是数学模式  
`array` 必须在数学模式中才能使用
- ☞ `eqnarray` 自动编号, `array` 不能
- ☞ `eqnarray` 每行为行间公式,  
`array` 为一个整体
- ☞ `eqnarray` 中间一列按行内公式选择字体大小,  
`array` 则全部按行内公式选择字体大小
- ☞ `eqnarray` 每行不能超过三列,  
`array` 可以任意多列
- ☞ `array` 可以指定每列的对齐方式,  
`eqnarray` 则不能

数学公式

潘建瑜

概述

行内公式与行间公式

常见数学公式排版

可以变大的定界符

数学字体和数学符号表

## 1 概述

## 2 行内公式与行间公式

## 3 常见数学公式排版

## 4 可以变大的定界符

## 5 数学字体和数学符号表



# 数学字体

数学公式

潘建瑜

概述

行内公式与行间公式

常见数学公式排版

可以变大的定界符

数学字体和数学符号表

## □ 数学字体命令

☞ `\mathrm`, `\mathit`, `\mathtt`

☞ `\mathsf`, `\mathbf`, `\mathcal`

☞ 其它更多数学字体需加载特殊宏包, 如:  $\mathbb{R}$

## □ 希腊字母: $\alpha, \beta, \gamma, \dots$ $\Gamma, \Omega, \Delta, \dots$

☞ 直立希腊字母可以使用 `pxfont` 宏包

## □ 数学粗体声明: `\boldmath`, `\unboldmath`

☞ 不能用于数学模式中, 可放在数学模式前

☞ `\bm` 宏包: `\bm{数学符号或公式}`

# 数学符号表

数学公式

潘建瑜

概述

行内公式与行间  
公式

常见数学公式排  
版

可以变大的定界  
符

数学字体和数学  
符号表

□ 二元运算符:  $\pm$ ,  $\mp$ ,  $\times$ ,  $\div$ ,  $\dots$

□ 关系运算符:  $\leq$ ,  $\geq$ ,  $\neq$ ,  $\subset$ ,  $\dots$

□ 箭头符号:  $\rightarrow$ ,  $\leftarrow$ ,  $\Rightarrow$ ,  $\Longleftrightarrow$ ,  $\dots$

□ 具有两种尺寸的符号:  $\Sigma$ ,  $\cup$   $\sum$ ,  $\cup$ ,  $\dots$

□ 其它符号:  $\forall$ ,  $\exists$ ,  $\clubsuit$ ,  $\spadesuit$ ,  $\dots$

□ 函数名:  $\sin$ ,  $\cos$ ,  $\lim$ ,  $\max$ ,  $\dots$

$\clubsuit$  更多数学符号见课程主页

# 第五章

## 常用文档类别与版式

潘建瑜

华东师范大学数学系

2010.03



# 本讲内容

- 1 文档类别及可选项
- 2 章节命令
- 3 文章标题
- 4 摘要
- 5 titlesec 宏包介绍

# 文档类别命令

```
\documentclass[可选项]{文档类别}
```

□ 常用标准文档: article, book, report, letter

- 👉 文档类别必须选一种, 且只能选一种
- 👉 不同的文档类别有不同的默认版式和组成部分
- 👉 book 类可含 chapter, 且对奇偶页采用不同的处理方式

# 可选项

- ❑ 可选项: 对整篇文档起作用, 可同时使用 **多个互不排斥** 的选项, 用逗号隔开

```
\documentclass[11pt,a4paper]{article}
```

- 👉 基本字体尺寸: 10pt, 11pt, 12pt
- 👉 纸张大小: letterpaper, a4paper, a5paper, ...
- 👉 排版方向: portrait (竖向), landscape (横向)
- 👉 标题页: titlepage, notitlepage
- 👉 final 和 draft

- 1 文档类别及可选项
- 2 章节命令
- 3 文章标题
- 4 摘要
- 5 titlesec 宏包介绍

# 章节命令

```
\章节命令[短标题]{标题}
```

```
\章节命令*{标题}
```

☞ 章节命令: `\part`, `\chapter`

`\section`, `\subsection`, `\subsubsection`,  
`\paragraph`, `\subparagraph`

☞ 短标题: 用于显示在目录和页眉, 缺省与标题相同

☞ 层次号: `\chapter` 为 0, `\section` 为 1, 向后逐次递增, `\part` 在 book 中为 -1, 在 article 中为 0



# 章节的自动编号

## □ book 类中的自动编号

☞ `\part`, `\chapter` 独立编号

☞ `\section`, `\subsection` 关联编号

## □ article 类中的自动编号

☞ `\part`, `\section` 独立编号

☞ `\subsection`, `\subsubsection` 关联编号

# 几点说明

♣ 带星号的章节命令不参加自动编号

♣ `\part` 和 `\chapter` 标题占两行

♣ 在 book 类中, `\part` 和 `\chapter` 只出现在奇数页, 且 `\part` 独占一页

🔧 标题中文化

```
\renewcommand{\partname}{}{}
```

```
\renewcommand{\thepart}{第~\Roman{part}~篇}
```

♣ 相类似地, 可以修改 `\chaptername`

# 人工修改自动编号

- 可根据需要修改自动编号所到达的最深层次

```
\setcounter{secnumdepth}{数}
```

- 👉 book 类: secnumberdepth 取值范围为 -2 至 5, 缺省为 2, 值为 -2 时取消所有编号
- 👉 article 类: secnumberdepth 取值范围为 -1 至 5, 缺省为 3, 值为 -1 时取消所有编号

# 章节计数器

♣ 自动编号的章节都有各自所对应的计数器:

part, chapter, section, subsection,  
subsubsection, paragraph, subparagraph

🔧 可以使用 `\setcounter` 来修改计数器的值

`\setcounter{计数器名}{数}`

🔧 每次调用不带 \* 号的章节命令时, 计数器的值自动加 1, 并将低层次的计数器置零

1 文档类别及可选项

2 章节命令

3 文章标题

4 摘要

5 titlesec 宏包介绍

# 文章标题

```
\title{标题}  
\author{标题}  
\date{标题}  
\thanks{标题}  
\maketitle
```

👉 L<sup>A</sup>T<sub>E</sub>X 用特定的字体和字号, 居中输出 标题

标题过长可以用 `\\` 强制换行

👉 若有多个作者, 可用 `\and` 隔开, 也可使用 `\\`

# 文章标题

☞ `\date` 可以省略, 此时自动加上当前日期, 例

`\date{2007年3月}`

`\date{}` → 不输出日期

☞ `\thanks` 可以出现在 `\title`, `\author` 和 `\date` 的参数中的任何位置, 也可出现多次

☞ `\maketitle` 用于输出以上标题信息, 不可省略

# 自定义标题页

```
\begin{titlepage}
```

标题页内容

```
\end{titlepage}
```

- ✎ 标题页内容 可以含有排版命令和文本, 可以排成任意想要的形式
- ✎ 该环境生成的标题页将独占一页
- ✎ 该环境重置页码计数器, 使得标题页的页码为 1



1 文档类别及可选项

2 章节命令

3 文章标题

4 摘要

5 titlesec 宏包介绍

# 摘要环境

```
\begin{abstract}  
    . . . . .  
\end{abstract}
```

📖 book 类没有 摘要

📖 摘要 标题中文化

```
\renewcommand{\abstractname}{摘\qqquad 要}
```

📖 例: 501book.tex, 502article.tex, 503.tex,  
503twocol.tex

- 1 文档类别及可选项
- 2 章节命令
- 3 文章标题
- 4 摘要
- 5 titlesec 宏包介绍

# titlesec 宏包

```
\usepackage{titlesec}
```

👉 设置章、节标题的格式

👉 主要命令: `\titleformat`, `\titlespace`

👉 例: 504book.tex



`\titlespacing{command}{left}{beforesep}{aftersep}`

☞ `command`: 需要设置的章节命令, 如:  
`\chapter`, `\section`, `\subsection`, ...

☞ `left`: 增加与左边界的间距

☞ `beforesep`: 增加标题上方的间距

☞ `aftersep`: 增加标题下方的间距

例: `\titlespacing{\section}{0pt}{3.5ex}{1.5\wordsep}`  
(`\wordsep`: inter-word space of the current font)

♣ `titlesec` 的详细使用参见其使用手册

# 第六章 图形

潘建瑜

华东师范大学数学系

2010.03



# 本讲内容

- ① 绘制图形
- ② 浮动的表格和插图



# 图形

## □ 绘制图像

$\text{\LaTeX}$  提供一些绘图命令, 可以绘制直线, 圆等一些简单的基本图形

## □ 插入图像

$\text{\LaTeX}$  可以插入外部图形, 如 eps 文件

♣ 参考“ $\text{\LaTeX} 2_{\epsilon}$  插图指南” (课程主页有下载)

□  $\text{\LaTeX}$  可调用许多绘图宏包来绘制复杂图形  
如: [Xy-pic](#), [PSTricks](#), [PGF/TikZ](#), [Asymptote](#)

# 1 绘制图形

## 2 浮动的表格和插图

# 图形与坐标系

## □ 一个图形总是与一个坐标系关联一起

☞ 单位长度: `\unitlength`, 缺省值为 1pt

重设单位长度可以放大和缩小所画的图形

☞ 坐标轴原点: 可放任意位置, 通常在图形左下角

## □ 图形环境: `picture`

```
\begin{picture}(宽, 高)(横坐标, 纵坐标)  
    ...  
\end{picture}
```

# 参数说明

- 👉 (宽, 高) → 指定图形所占区域(图形盒子)的大小
- 👉 图形盒子 → 深度为 0, 基准点为左下角
- 👉 (横坐标, 纵坐标) → 图形盒子基准点的坐标  
改变基准点的坐标就可以平移图形  
可以省略该选项, 此时基准点为坐标轴原点

# 参数说明

- 👉 (宽, 高) → 指定图形所占区域(图形盒子)的大小
- 👉 图形盒子 → 深度为 0, 基准点为左下角
- 👉 (横坐标, 纵坐标) → 图形盒子基准点的坐标  
改变基准点的坐标就可以平移图形  
可以省略该选项, 此时基准点为坐标轴原点

## □ 确定图形元素在坐标中的位置

```
\put(x, y){图形元素}
```

```
\multiput(x, y)(\Delta x, \Delta y){个数}{图形元素}
```

- 👉 图形元素可以是普通图形, 文字或子图等

# 直线

## □ 绘制直线

```
\line( $\Delta x$ ,  $\Delta y$ ){长度}
```

☞ 长度以 `\unitlength` 为单位

斜线的长度是指其在横轴上的投影的长度

☞  $(\Delta x, \Delta y)$  确定直线的方向数

- 取值只能为  $0, \pm 1, \pm 2, \pm 3, \pm 4, \pm 5, \pm 6$

- $\Delta x, \Delta y$  不能有公因子

☞ 斜线的长度不能小于 10pt 或 3.5mm

# 矢量线

## □ 绘制矢量线

```
\vector( $\Delta x$ ,  $\Delta y$ ){长度}
```

☞ 长度以 `\unitlength` 为单位

斜线的长度是指其在横轴上的投影的长度

☞  $(\Delta x, \Delta y)$  取值只能为  $0, \pm 1, \pm 2, \pm 3, \pm 4$

☞  $\Delta x, \Delta y$  不能有公因子

☞ 斜矢量线的 长度 不能小于 10pt 或 3.5mm

# 线的粗细

## □ 指定直线或曲线粗细的声明

👉 `\thinlines` → 缺省值

👉 `\thicklines`

👉 `\linethickness{粗细}`

♣ `\linethickness` 只对水平线和竖线起作用

👉 例: 6-2-1.tex



# 圆

## □ 绘制圆

👉 `\circle{直径}` → 圆周

👉 `\circle*{直径}` → 实心圆

♣ 圆由小圆弧组成, 由于事先设计好的小圆弧只有有限多种,  $\text{\LaTeX}$  只能画特定尺寸的圆, 圆周直径不能超过 40pt, 实心圆直径不能超过 15pt

👉 画圆时定位命令 `\put` 或 `\multiput` 中的坐标是圆心的坐标

👉 例: 6-2-2.tex

# 圆角矩形

## 圆角矩形: 矩形的四角为四分之一圆

```
\oval(宽, 高)[部分]
```

### 可选参数 部分 的取值:

☞ **t** → 上半圆角矩形, **b** → 下半圆角矩形

**l** → 左半圆角矩形, **r** → 右半圆角矩形

☞ **tl** → 左上角四分之一圆角矩形

**tr** → 右上角四分之一圆角矩形

**bl** → 左下角四分之一圆角矩形

**br** → 右下角四分之一圆角矩形

☞ 例: 6-2-3.tex

# 图形中的盒子

```
\makebox(宽, 高)[位置]{...}  
\framebox(宽, 高)[位置]{...}  
\dashbox{虚线长度}(宽, 高)[位置]{...}
```

## 参数说明:

- ➡ 宽, 高和虚线长度 以 `\unitlength` 的值为单位
- ➡ 盒子的基准点为左下角
- ➡ 定位命令 `\put` 或 `\multiput` 中的坐标是盒子的基准点

# 参数说明

 **位置**: 指定文本在盒子中的位置


**t** → 水平居中, 竖直**顶端**, **b** → 水平居中, 竖直**底端**

**l** → 水平**居左**, 竖直居中, **r** → 水平**居右**, 竖直居中

**tl** → 左上角, **tr** → 右上角

**bl** → 左下角, **br** → 右下角

**s** → 竖直居中, 水平伸展充满盒子

 例: 6-2-4.tex

# 图形中的文本

```
\put(x,y){...}
```

```
\put(x,y){\makebox(0,0)[位置]{...}}
```

```
\put(x,y){\shortstack[位置]{...}}
```

- ☞ 第一个命令中的文本不会换行
- ☞ 第三个命令用于竖排文本 (利用强制换行符)
- ☞ 盒子的基准点为左下角
- ☞ 定位命令 `\put` 或 `\multiput` 中的坐标是盒子的基准点
- ☞ 例: 6-2-5.tex

# Bézier 曲线

```
\bezier{n}(x_1, y_1)(x_2, y_2)(x_3, y_3)  
\qbezier[n](x_1, y_1)(x_2, y_2)(x_3, y_3)
```

- 用  $n + 1$  个点绘制一条从  $(x_1, y_1)$  到  $(x_3, y_3)$  的 Bézier 曲线
  - 👉  $(x_2, y_2)$  用于控制曲线的弯曲程度
  - 👉 推荐使用第二条命令
  - 👉 例: 6-2-6.tex

# 子图

## 子图: 图形中的部分图形作为一个整体看待

👉 picture 环境的嵌套

👉 子图有自己的坐标系

可以自己设置单位长度和线的粗细

👉 子图可以放在一个盒子里, 重复使用

👉 例: 6-3-1.tex, 6-3-2.tex, 6-3-3.tex

```
\newsavebox{子图盒子名}
```

```
\savebox{子图盒子名}(宽,高)[位置]{子图}
```

```
\usebox{子图盒子名}
```

```
\sbox{子图盒子名}{}
```

1 绘制图形

2 浮动的表格和插图



# 浮动的图表

## ❑ 浮动的图表

- 👉 `tabular` 和 `picture` 环境生成的图表通常插入在当前位置, 如果当前页的剩余空间不够时, 会造成排版出现很大的空白
- 👉 浮动图表功能: 根据需要自动调整图表的位置

## ❑ 浮动图表环境:

- 👉 浮动的图形: `figure` 和 `figure*`
- 👉 浮动的表格: `table` 和 `table*`

(例: 651.tex)

# 浮动图表环境

## ❑ 浮动的图形环境

<code>\begin{figure}[位置]</code>	<code>\begin{figure*}[位置]</code>
<code>...</code>	<code>...</code>
<code>\end{figure}</code>	<code>\end{figure*}</code>

## ❑ 浮动的表格环境

<code>\begin{table}[位置]</code>	<code>\begin{table*}[位置]</code>
<code>...</code>	<code>...</code>
<code>\end{table}</code>	<code>\end{table*}</code>

# 浮动图表环境

- 带星号的图表环境只适用于双列页面格式, 此时图表占据两列
- 可选项 **位置** 由零个到四个字母组成:  
 $h \rightarrow \text{here}, t \rightarrow \text{top}, b \rightarrow \text{bottom}, p \rightarrow \text{page}$   
 若在字母前加 **!**, 表示去掉所加的限制  
 缺省值为 **tbp**, 字母顺序无关紧要:  $h \rightarrow t \rightarrow b \rightarrow p$
- L<sup>A</sup>T<sub>E</sub>X** 按照**先进先出**的顺序排版浮动表格
- \clearpage**  $\rightarrow$  强制排版浮动队列中的所有剩余对象, 并开始新的一页
- float** 宏包: 提供 **H** 选项, 强制在当前位置输出

# 浮动图表编号

## □ 浮动图形和表格的自动编号

👉 在浮动图表环境中使用 `\caption` 命令

`\caption[短标题]{标题\label{引用标志}}`

添加标题并自动编号

👉 插图和表格单独编号

## □ 排印插图目录和表格目录

👉 `\listoffigures`

👉 `\listoftables`

# 浮动图表编号前缀的中文化

```
\renewcommand*\figurename{图}  
\renewcommand*\tablename{表}  
  
\renewcommand*\listfigurename{插图目录}  
\renewcommand*\listtablename{表格目录}
```

# 第七章 自定义与改错

潘建瑜

华东师范大学数学系

2010.04



# 本讲内容

- 1 自定义命令
- 2 计数器
- 3 长度
- 4 出错信息与警告信息

# 自定义命令

```
\newcommand{新命令}[参数个数]{原命令}
```

```
\newcommand*{新命令}[参数个数]{原命令}
```

```
\renewcommand{新命令}[参数个数]{原命令}
```

```
\renewcommand*{新命令}[参数个数]{原命令}
```

☞ 参数个数不能超过 9 个

☞ 带星号的称为“**短命令**” → 不能出现段落

☞ L<sup>A</sup>T<sub>E</sub>X 编译时用**原命令**内容替换**新命令**, 同时将相应的参数代入

❖ 自定义命令本质上就是**命令替换**



# 自定义命令举例

```
\newcommand{\eps}{\epsilon}
```

```
\newcommand{\eps}{\epsilon}
```

```
\newcommand{\eps}{\ensuremath{\epsilon}}
```

 `\ensuremath`

→ 使新命令在数学模式内外都可以使用

```
\newcommand{\anvec}[2]{\{#1\}_1, \ldots, \{#1\}_{#2}}
```

 若新定义的命令中含参数, 最好都加上大括弧

# 自定义命令的作用范围

## 自定义命令的作用范围

- 如果放在**导言区**, 则全局有效
- 如果在正文中的某个环境或分组中定义, 则仅在其所在环境或分组内部有效
- 例: 701.tex

## 1 自定义命令

## 2 计数器

## 3 长度

## 4 出错信息与警告信息

## ❑ 常见内部计数器

part, chapter, section, subsection, equation,  
page, footnote, ...

## ❑ 内部计数器的从属关系

part → chapter → section → subsection

## ❑ 计数器的关联

```
\numberwithin{从计数器}{主计数器}
```

## ❑ 自定义计数器

```
\newcounter{新计数器名}[上级计数器]
```

👉 上级计数器 是可选项

## ❑ 修改计数器的值

☞ `\setcounter{计数器名}{数}`

→ 给指定的计数器赋值

☞ `\addtocounter{计数器名}{数}`

→ 给指定的计数器加上一个值

☞ `\stepcounter{计数器名}`

→ 将指定的计数器的值加 1

→ 同时将所有下属计数器的值置零

## ❑ 取计数器的值

`\value{计数器名}`

## ❑ 设置计数器的值的显示方式

`\数字样式命令{计数器名}`

☞ 数字样式命令 有:

`\arabic`, `\roman`, `\Roman`, `\alph`, `\Alph`, `\fnsymbol`

☞ `\arabic` : 阿拉伯数字 (1, 2, 3, 4, ...)

☞ `\roman`, `\Roman` : 罗马数字  
(i, ii, iii, iv, ..., I, II, III, IV, ...)

☞ `\alph`, `\Alph` : 拉丁字母, 不能超过 26  
(a, b, c, ..., A, B, C, ...)

☞ `\fnsymbol` : 脚注标号, 不能超过 9  
(\*, †, ‡, §, ¶, ||, \*\*, ††, ‡‡)

## ❑ 使用计数器的值: \the计数器名

```
\renewcommand{\thesection}{\S\arabic{section}}
```

👉 例: 修改公式编号的显示方式

```
\renewcommand{\theequation}%  
{\thesection.\arabic{equation}}
```

```
\numberwithin{equation}{section}
```

## 1 自定义命令

## 2 计数器

## 3 长度

## 4 出错信息与警告信息



## ❑ 常见的长度命令:

`\textwidth`, `\textheight`, `\baselineskip`  
`\parindent`, `\parskip`, ...

## ❑ 修改长度的值:

```
\setlength{\长度命令}{长度值}
\addtolength{\长度命令}{长度值}
```

- 👉 **长度值** 须带长度单位, 或者是已定义长度的倍数
- 👉 **长度值** 中可以使用弹性长度

## ❑ 自定义新长度命令

```
\newlength{\新长度命令}
```

## ❑ 将长度命令的值设为指定文本的宽度, 高度和深度

```
\settowidth{\长度命令}{文本}  
\settoheight{\长度命令}{文本}  
\settodepth{\长度命令}{文本}
```

## ❑ 编译时显示长度命令的值: \showthe\长度命令

👉 这个值不会出现在排版结果中

## 1 自定义命令

## 2 计数器

## 3 长度

## 4 出错信息与警告信息

# 出错信息

- ❑ 出错信息通常来自  $\text{\LaTeX}$  本身 或 底层的  $\text{\TeX}$
- ❑ 出错的处理方法

# 警告信息

- ❑ Overfull
- ❑ Underfull
- ❑ There were undefined references
- ❑ There were multiply-defined labels
- ❑ . . . .

# 第八章

# 文字模式高级技巧

潘建瑜

华东师范大学数学系

2010.04

# 本讲内容

- ① 文本位置
- ② 引文与抄录
- ③ 盒子
- ④ 表格与罗列
- ⑤ 脚注与边注
- ⑥ 段落形状

# 文本位置

## ❑ 文本居左环境: `flushleft`

```
\begin{flushleft}  
...  
\end{flushleft}
```

## ❑ 文本居左声明

```
\raggedright
```

👉 例: 801.tex



# 文本位置

## ❑ 文本居右环境: `flushright`

```
\begin{flushright}
    ...
\end{flushright}
```

## ❑ 文本居右声明

```
\raggedleft
```

👉 与 `center` 环境一样, L<sup>A</sup>T<sub>E</sub>X 会在这些环境的前后插入额外的垂直间隔

## ① 文本位置

## ② 引文与抄录

## ③ 盒子

## ④ 表格与罗列

## ⑤ 脚注与边注

## ⑥ 段落形状

# 引文与抄录

## □ 引文环境: quotation 和 quote

☞ 引文: 另起一段, 并在两边向内缩进一定距离

☞ 引文可以是一行, 或多行, 甚至几个段落

## □ quotation 和 quote 的区别

☞ quotation 中的段落与正常段落一样

☞ quote 中的段落首行不自动缩进, 但增加段间距

## □ 西文诗歌, 韵文环境: verse

☞ 引文, 诗歌环境可以嵌套, 但嵌套层数  $\leq 6$

## ❑ 抄录环境: `verbatim` 和 `verbatim*`

- 👉 抄录: 按原样输出文本
- 👉 所有的  $\text{\LaTeX}$  命令都被作为普通字符串输出
- 👉 西文用打字机字体, 汉字为进入该环境前的字体
- 👉 带星号的环境用 `\` 显示空格

## ❑ 抄录命令: `\verb` 和 `\verb*`

- 👉 适用于不超过一行的文本
- 👉 带星号的命令用 `\` 显示空格

## ❏ 抄录命令使用格式

```
\verb|抄录文本|  
\verb*|抄录文本|
```

- 👉 抄录命令中的**抄录文本**不能多于一行
- 👉 定界符 | → 可以使用除星号和空格以外的任何**抄录文本**中没有出现的符号
- 👉 左右定界符必须一致
- 👉 反引号和单引号在抄录环境中分别显示为 ‘ 和 ’

## ① 文本位置

## ② 引文与抄录

## ③ 盒子

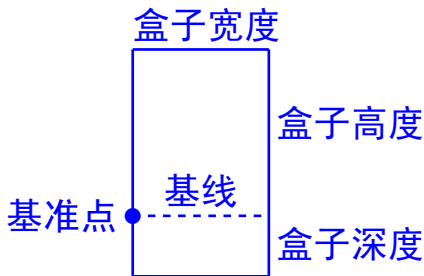
## ④ 表格与罗列

## ⑤ 脚注与边注

## ⑥ 段落形状

# 盒子

□ 在  $\text{\LaTeX}$  中, 一切都是盒子!



👉  $\text{\LaTeX}$  的排版过程就是在构造盒子和堆砌盒子

👉 盒子的基准点通常也称为参考点

# 盒子

- 👉 当盒子左右排列时, 它们的**基线**在同一水平线上
- 👉 当盒子上下排列时, **基准点**在同一垂直线上
- 👉  $\backslash width$  → 表示盒子的宽度
- $\backslash height$  → 表示盒子的高度
- $\backslash depth$  → 表示盒子的深度
- $\backslash totalheight$  → 表示盒子的**总高度**
- 👉  $\backslash totalheight = \backslash height + \backslash depth$



# 使用命令构造盒子

## ❑ LR 盒子

包含的内容只能位于同一行, 不能被分成多行

## ❑ 段落盒子

右垂直堆叠的行构成, 即可以包含多行文本

## ❑ 标尺盒子

为一个实心矩形, 通常用于画水平线或垂直线

# LR 盒子

## ❑ 创建 LR 盒子

```
\mbox{单行文本}  
\fbox{单行文本}  
\makebox[宽度][位置]{单行文本}  
\framebox[宽度][位置]{单行文本}
```

- 🔔 单行文本: 不能分行, 但可以是一个环境或盒子
- 🔔 `\fbox` 和 `\framebox` 生成带框的盒子
- 🔔 前两个命令生成的盒子宽度等于文本的实际宽度 (自然宽度)

# LR 盒子

- 👉 后两个命令可以指定盒子的宽度和文本排放位置  
`l`, `r`, `s`, 缺省为居中
- 👉 后两个命令的选项省略一个时, 只能省略位置
- 👉 当指定的盒子宽度小于文本自然宽度时,  
文本将会超出盒子
- 👉 盒子宽度为 0 时的特殊作用. 例 802.tex
- 👉 盒子的宽度可以用盒子的自然尺寸来定义
- 👉 盒子框线与文本之间的距离: `\fboxsep`  
盒子框线的粗细: `\fboxrule`  
上面的两个长度可以使用 `\setlength` 进行修改

# 盒子的多次使用

❶ 创建新盒子: `\newsavebox{盒子名}`

❷ 将文本保存到盒子中

```
\sbox{\盒子名}{文本}  
\savebox{\盒子名}[宽度][位置]{文本}
```

❸ 使用盒子: `\usebox{盒子名}`

👉 给盒子加框: `\fbox{\usebox{盒子名}}`

👉 排版时,  $\text{\LaTeX}$  只按盒子的尺寸安排所占空间

# 盒子的多次使用

- 👉 如果需要保存大段文本到一个盒子里，  
可以使用 `lrbox` 环境

```
\begin{lrbox}{\盒子名}  
...  
\end{lrbox}
```

该环境的作用与 `\sbox` 一样

# LR 盒子的升降

❑ 盒子的升降: 在竖直方向移动一定的距离

```
\raisebox{升高量}[盒高][盒深]{文本}
```

- 👉 该命令生成一个无框的盒子, 盒子的基线位于当前行的基线上, 但盒子内文本的基线相对于盒子的基线在竖直方向作一定的移位
- 👉 升高量为正 → 将盒子上移  
升高量为负 → 将盒子下移
- 👉 选项盒高和盒深缺省时, 自动调整大小
- 👉 若升高量过大, 文本可能会超出盒子的范围
- 👉 盒高和盒深的值为负时, 被当成 0 处理

# 关于盒子的几点说明

## □ 关于盒子的几点说明

- ☞ 同一行中盒子的基线位于同一水平线上, 这些盒子组成一个更大的盒子, 大盒子的高度和深度分别为各小盒子高度和深度的最大值
- ☞ 高度和深度为 0 的升降盒子对于排版一些特殊的表格或矩阵非常有用.

# 标尺盒子

□ 标尺盒子: 一个实心矩形

```
\rule[升高量]{水平宽度}{竖直宽度}
```

- ☞ 可选项升高量指定标尺盒子底部相对于当前行基线向上移动的距离
- ☞ 宽度为 0 的标尺盒子是看不见的, 但可起“支撑”作用.
- ☞ 例: 802.tex



# 子段盒子与小页环境

## ❑ 将整段文本放入一个盒子的两种实现方法:

### ① 子段盒子命令:

```
\parbox[位置]{宽度}{文本}
```

### ② 小页环境:

```
\begin{minipage}[位置]{宽度}  
...  
\end{minipage}
```



👉 以上两种方法都生成一个指定宽度的段落盒子

👉 排版时, 盒子被看成一个不可分割的整体

# 子段盒子与小页环境

```
\parbox[位置]{宽度}{文本}
```

```
\begin{minipage}[位置]{宽度}  
  ...  
\end{minipage}
```

-  **位置**用来指定盒子与外部文本的对齐方式:
- b** → 盒子底行文本基线与盒外基线对齐
  - t** → 盒子顶行文本基线与盒外基线对齐
  - 缺省为垂直居中, 即盒子中部与盒外基线对齐
-  盒子的总高度缺省为所含内容的自然高度

# 子段盒子与小页环境

```
\parbox[位置][总高度][内部位置]{宽度}{文本}
```

```
\begin{minipage}[位置][总高度][内部位置]{宽度}  
\end{minipage}
```

- ☞ 可以显式指定盒子的总高度
- 可选项内部位置可以为: t, b, c, s
- 可选项总高度和内部位置应同时使用或同时不用

## ① 文本位置

## ② 引文与抄录

## ③ 盒子

## ④ 表格与罗列

## ⑤ 脚注与边注

## ⑥ 段落形状

# 表格环境

## ❑ 表格环境: tabular 和 tabular\*

```
\begin{tabular}[竖向位置]{列格式}
...
\end{tabular}
```

```
\begin{tabular*}{宽度}[竖向位置]{列格式}
...
\end{tabular*}
```

👉 **tabular\*** 环境可以指定整个表格的宽度

👉 竖向位置: **t**, **b**

👉 列格式参数: **l**, **c**, **r**, **|**, **||**

## □ 更多列格式参数:

☞  $p\{\text{宽度}\}$  → 指定对应列的宽度, 居左, 自动换行

☞  $\ast\{\text{数}\}\{\text{列格式}\}$  → 多个相同列格式的缩写

## □ 更多列格式参数:

- 👉  $p\{\text{宽度}\}$  → 指定对应列的宽度, 居左, 自动换行
- 👉  $\ast\{\text{数}\}\{\text{列格式}\}$  → 多个相同列格式的缩写

## □ 边界和列间距相关

- 👉  $@\{\text{文本}\}$  → @表达式, 在对应位置“吃掉”原有的列间隔, 改为插入指定的文本
- 👉 若只写  $@\{\}$ , 则表示取消对应位置的列间隔
- 👉 若要改变指定相邻两列的列间距, 可在 @表达式中使用  $\backslash\text{hspace}$
- 👉 若在 @表达式中使用  $\backslash\text{extracolsep}\{\text{宽度}\}$  命令, 则其后面所有的列间距都增加指定的额外长度

## ❑ 控制表格式样的常用参数

☞ `\arrayrulewidth` → 各类表格中线条的粗细

☞ `\doublerulesep` → 双线之间的距离

☞ `\arraystretch` → 各类表格中行间距伸缩因子

☞ `\tabcolsep` → 表格环境中列间距的一半

☞ `\arraycolsep` → `array` 环境中列间距的一半



## □ 控制表格式样的常用参数

☞ `\arrayrulewidth` → 各类表格中线条的粗细

☞ `\doublerulesep` → 双线之间的距离

☞ `\arraystretch` → 各类表格中行间距伸缩因子

☞ `\tabcolsep` → 表格环境中列间距的一半

☞ `\arraycolsep` → `array` 环境中列间距的一半

→ 可使用 `\setlength` 或 `\renewcommand` 修改这些参数的值

→ 例: 803.tex

# 罗列环境

## □ 常用的三种罗列环境

- ☞ `itemize` → 带相同的标签的罗列环境
- ☞ `enumerate` → 条目标签为自动编号的符号
- ☞ `description` → 手动指定各条目的标签

# 罗列环境

## ❑ 常用的三种罗列环境

- ☞ `itemize` → 带相同的标签的罗列环境
- ☞ `enumerate` → 条目标签为自动编号的符号
- ☞ `description` → 手动指定各条目的标签

## ❑ 条目输入格式

`\item[标签] 条目内容`

- 前两种罗列环境可以省略标签, 第三种不能省

# 罗列环境

## ❑ 常用的三种罗列环境

- 👉 `itemize` → 带相同的标签的罗列环境
- 👉 `enumerate` → 条目标签为自动编号的符号
- 👉 `description` → 手动指定各条目的标签

## ❑ 条目输入格式

`\item[标签] 条目内容`

- 前两种罗列环境可以省略标签, 第三种不能省
- 罗列可以相互嵌套, 但每一种罗列的嵌套层数  $\leq 4$

# itemize 罗列环境

## ❑ itemize 罗列环境



```
\begin{itemize}
  \item[标签] 条目内容
  ...
  \item[标签] 条目内容
\end{itemize}
```

- 👉 标签是可选项, 若指定, 则使用指定的标签
- 👉 缺省的标签与层数有关, 分别为:  $\bullet$ ,  $-$ ,  $*$ ,  $\cdot$
- 👉 各层标签所对应的命令: `\labelitemi`,  
`\labelitemii`, `\labelitemiii`, `\labelitemiv`
- ➔ 可以使用命令 `\renewcommand` 修改

# enumerate 罗列环境

## enumerate 罗列环境

```
\begin{enumerate}  
  \item[标签] 条目内容  
  ...  
  \item[标签] 条目内容  
\end{enumerate}
```

-  标签是可选项, 若指定, 则使用指定的标签
-  缺省标签为自动编号的符号, 与层数有关, 分别为:
  - ① 阿拉伯数字后跟圆点: 1. 2.
  - ② 圆括号包围的小写拉丁字母: (a) (b)
  - ③ 小写罗马数字后跟圆点: i. ii.
  - ④ 大写拉丁字母后跟圆点: A. B.

# enumerate 罗列环境


```
\begin{enumerate}  
  \item[标签] 条目内容  
  ...  
  \item[标签] 条目内容  
\end{enumerate}
```

- 👉 各层标签所对应的命令: `\labelenumi`,  
`\labelenumii`, `\labelenumiii`, `\labelenumiv`
- 👉 各层缺省标签所对应的计数器为:  
`enumi`, `enumii`, `enumiii`, `enumiv`
- 👉 可以指定计数器的值的显示方式:  
`\arabic`, `\roman`, `\Roman`, `\alph`, `\Alph`

# description 罗列环境

## description 罗列环境

```
\begin{description}
  \item[标签] 条目内容
  ...
  \item[标签] 条目内容
\end{description}
```

 该罗列环境没有指定缺省的标签, 故**标签**选项一定要给出



# 广义罗列环境

## □ 广义罗列环境: list

```
\begin{list}{标准标签}{罗列声明}  
  \item[标签] 条目内容  
  ...  
  \item[标签] 条目内容  
\end{list}
```

👉 标准标签 → 指定缺省标签

👉 罗列声明 → 指定罗列的样式, 包括:  
标签类型与宽度, 条目缩进距离, 段落间距, ...

# 广义罗列环境样式参数

## ❑ 广义罗列环境中的样式参数

☞ `\topsep`, `\partopsep`, `\itemsep`, `\parsep`

☞ `\labelwidth`, `\labelsep`

☞ `\itemindent`, `\listparindent`

☞ `\leftmargin`, `\rightmargin`

# 广义罗列环境样式参数

## □ 广义罗列环境中的样式参数

☞ `\topsep`, `\partopsep`, `\itemsep`, `\parsep`

☞ `\labelwidth`, `\labelsep`

☞ `\itemindent`, `\listparindent`

☞ `\leftmargin`, `\rightmargin`

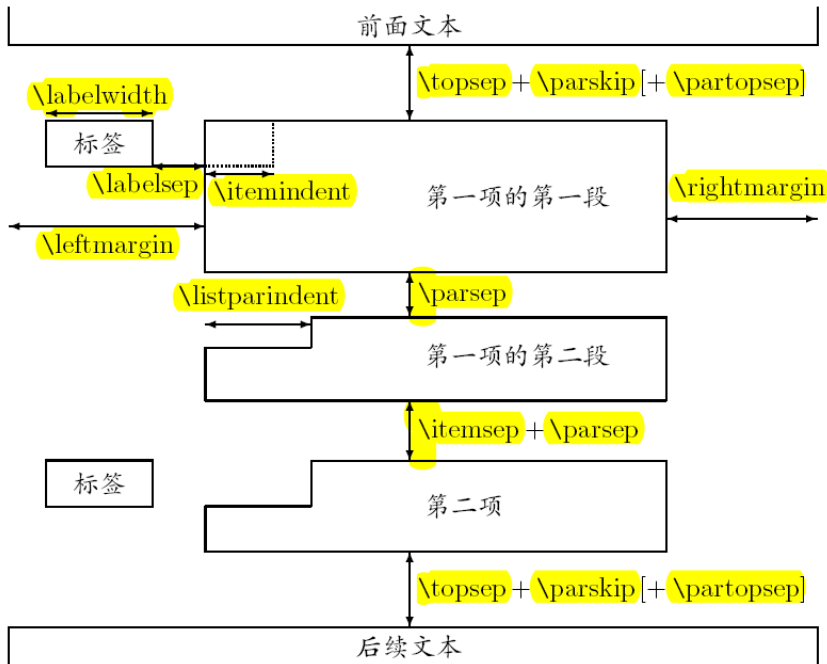
→ 可使用 `\setlength` 等命令修改以上参数的值

→ 标签盒子缩进距离:

`\leftmargin`—`\labelwidth`—`\labelsep`

→ 若要使用自动编号的标签, 则需要使用计数器:

`\newcounter`, `\usecounter` (例: 804.tex)



# 平凡罗列环境

## □ 平凡罗列环境: `trivlist`

```
\begin{trivlist}
  \item[标签] 条目内容
  ...
  \item[标签] 条目内容
\end{trivlist}
```

☞ 主要用于生成其它环境, 如:  
`center`, `flushleft`, `flushright`

## ① 文本位置

## ② 引文与抄录

## ③ 盒子

## ④ 表格与罗列

## ⑤ 脚注与边注

## ⑥ 段落形状

# 脚注

## □ 脚注命令

`\footnote{脚注文本}` → 自动编号

`\footnote[数]{脚注文本}` → 指定脚注标记\*

\* 指定标记的脚注不参加自动编号

# 脚注

## 脚注命令

`\footnote{脚注文本}` → 自动编号

`\footnote[数]{脚注文本}` → 指定脚注标记\*

\* 指定标记的脚注不参加自动编号

👉 对应的计数器为 `footnote`

👉 修改脚注编号的显示样式

```
\renewcommand{\thefootnote}{\fnsymbol{footnote}}
\renewcommand{\thefootnote}{\arabic{footnote}}
```



# 脚注

## 脚注命令

`\footnote{脚注文本}` → 自动编号

`\footnote[数]{脚注文本}` → 指定脚注标记\*

\* 指定标记的脚注不参加自动编号

👉 对应的计数器为 `footnote`

👉 修改脚注编号的显示样式

```
\renewcommand{\thefootnote}{\fnsymbol{footnote}}  
\renewcommand{\thefootnote}{\arabic{footnote}}
```

👉 小页环境有独立的脚注计数器 `mpfootnote`  
缺省的显示样式为小写拉丁字母

# 禁止模式中的脚注

❑ 禁止模式: 数学模式, 表格, LR 盒子, 子段盒子

# 禁止模式中的脚注

❑ 禁止模式: 数学模式, 表格, LR 盒子, 子段盒子

❑ 在禁止模式中使用脚注

🔑 将脚注命令分成两部分:

`\footnotemark` → 写脚注标记命令

`\footnotetext{脚注文本}` → 写脚注文本命令

`\footnotemark[数]`

`\footnotetext[数]{脚注文本}`

# 禁止模式中的脚注

❑ 禁止模式: 数学模式, 表格, LR 盒子, 子段盒子

❑ 在禁止模式中使用脚注

🗨️ 将脚注命令分成两部分:

`\footnotemark` → 写脚注标记命令

`\footnotetext{脚注文本}` → 写脚注文本命令

`\footnotemark[数]`

`\footnotetext[数]{脚注文本}`

➔ 脚注标记命令位于禁止模式中需要注释的地方  
脚注文本命令则位于禁止模式之外 (例:  
805.tex)

# 禁止模式中的脚注

## ❑ 脚注标记与脚注文本的匹配

👉 指定脚注标记: 同一脚注必须使用统一的标记

👉 脚注自动编号时:

- 每调用一次脚注标记命令, `footnote` 的值就加 1
- 脚注文本命令只使用 `footnote` 值, 不改变其值
- 禁止模式中含多个脚注时, 在禁止模式外使用脚注文本命令时需手动调整 `footnote` 值, 同时每用一次脚注文本命令后, 需增加脚注计数器的值

```
\addtocounter{footnote}{负数}  
\stepcounter{footnote}
```

# 边注

## □ 边注: 在页面边上添加注释

```
\marginpar{边注文本}
```

- 👉 边注通常为一个较窄的字段盒子 → 边注盒子
- 👉 边注默认出现在页面右侧
- 👉 若使用双面选项(`twoside`), 则奇右偶左
- 👉 若使用双列选项(`twocolumn`), 则左左右右
- 👉 边注内容左对齐, 若要右对齐需用 `\flushright`

# 边注

## □ 边注: 在页面边上添加注释

```
\marginpar{边注文本}
```

- 👉 边注通常为一个较窄的字段盒子 → 边注盒子
- 👉 边注默认出现在页面右侧
- 👉 若使用双面选项(`twoside`), 则奇右偶左
- 👉 若使用双列选项(`twocolumn`), 则左左右右
- 👉 边注内容左对齐, 若要右对齐需用 `\flushright`
- 改变边注的默认位置: `\reversemarginpar`
- 例: 805.tex

# 边注

## ❑ book 类边注的处理方法:

```
\marginpar[\flushright 左边边注]{右边边注}
```

- 👉 若边注出现在奇数页, 则输出右边注, 并位于右边  
若边注出现在偶数页, 则输出左边注, 并位于左边

## ❑ 控制边注样式的参数

```
\marginparwidth → 边注盒子宽度
```

```
\marginparsep → 边注盒子与正文之间的距离
```

```
\marginparpush → 两个边注间的最小竖直间距
```



## ① 文本位置

## ② 引文与抄录

## ③ 盒子

## ④ 表格与罗列

## ⑤ 脚注与边注

## ⑥ 段落形状

# 段落形状

## □ 使用 $\text{T}_\text{E}\text{X}$ 命令改变段落形状

👉 移动段落边界:

```
\leftskip=长度  
\rightskip=长度
```

👉 多行缩进 (多行悬挂):

```
\hangafter=整数 → 指定缩进的行  
\hangindent=长度 → 指定缩进距离
```

→  $\text{\texttt{\textbackslash hangafter}}=n$  表示从第  $n + 1$  行开始缩进  
若  $n < 0$ , 则表示前面的  $|n|$  行缩进 (例: 806.tex)

→  $\text{\texttt{\textbackslash hangindent}}$  为正时表示左缩进, 为负时表示右缩进

# 段落形状

## □ 使用 $\text{T}_\text{E}\text{X}$ 命令改变段落形状

👉 段落形状:

$\backslash\text{parshape} = n \ i_1 \ l_1 \ i_2 \ l_2 \ \cdots \ i_n \ l_n$

- $n$  表示控制的行数
- $i_k, l_k$  成对出现, 分别表示第  $k$  行的左缩进距离和该行的长度

# 第九章

# 数学公式排版技巧

## 第一部分

潘建瑜

华东师范大学数学系

2010.04

# 本讲内容

- ① 数学排版的国际标准
- ② 数学模式中的参数
- ③ 定理定义的排版
- ④ 巧妙使用阵列环境
- ⑤ 多行公式左列问题

# ISO 数学排版公认标准

- ❑ 用粗斜体表示向量, 如:  $B, u, \omega$
- ❑ 用无衬线斜体表示矩阵与 2 阶张量, 如:  $M, D, I$
- ❑ 特殊常数与微分算子用直立体, 如:  $e, i, d$
- ❑ 数与单位构成的度量是一个整体, 不可分割  
数与单位之间留一小间隙, 单位用直立体  
如:  $5.3\text{ km}, 62\text{ kg}$

# 数学模式中的字体尺寸

## □ 数学模式中的四种字体尺寸

`\displaystyle` → 简记为 D

`\textstyle` → 简记为 T

`\scriptstyle` → 简记为 S

`\scriptscriptstyle` → 简记为 SS

- ☞ 对于变量, D 尺寸和 T 尺寸是一样的
- ☞ 数学模式中的变量和数学符号等自动选择尺寸
- ☞ `eqnarray` 中基本字体尺寸为 D
- ☞ `array` 中基本字体尺寸为 T

## ① 数学排版的国际标准

## ② 数学模式中的参数

## ③ 定理定义的排版

## ④ 巧妙使用阵列环境

## ⑤ 多行公式左列问题



# 数学模式中的参数

## □ 数学模式中的参数

☞ `\arraycolsep`

→ `array/eqnarray` 列间距的一半, 缺省为 5.0pt

☞ `\jot` → `eqnarray` 中的行间隔, 缺省为 3.0pt

☞ `\abovedisplayskip`

→ 长行间公式与上方文本的间隔

→ 缺省为 8.5pt plus 3pt minus 4pt

☞ `\belowdisplayskip`

→ 长行间公式与下方文本的间隔

→ 缺省为 8.5pt plus 3pt minus 4pt

# 数学模式中的参数

## □ 数学模式中的参数(续)

☞ `\abovedisplayshortskip`

- 短行间公式与上方文本的间隔
- 缺省为 0pt plus 2pt

☞ `\belowdisplayshortskip`

- 短行间公式与下方文本的间隔
- 缺省为 4pt plus 2pt minus 2pt

☞ `\mathindent`

- 使用 `fleqn` 选项时, 行间公式的缩进量
- 缺省为 25pt

# 长行间公式与短行间公式

- 👉 行间公式的“长”与“短”并不是指公式本身的长短，而是与其上方的文本相比较而言
- 👉 如果行间公式上方文本的结尾处在公式左边界的左边，则该公式为短公式，否则为长公式
- 👉 例 901.tex
- 👉 多行环境中的公式均是长公式  
如：`eqnarray`, `eqnarray*`
- ➔ 输入行间公式时，数学环境中不能有空行！

## ① 数学排版的国际标准

## ② 数学模式中的参数

## ③ 定理定义的排版

## ④ 巧妙使用阵列环境

## ⑤ 多行公式左列问题

# 定理类环境

□ **定理类环境**: 定理, 引理, 推论, 命题, 定义等

☞ 需要用特定的格式显示

☞ 带有特定的标题

☞ 自动编号

**Theorem 9.1** (Fermat). *There do not exist integers  $x, y, z$  and  $n > 2$  such that  $x^n + y^n = z^n$ .*

# 定理类环境的实现

## □ 定理类环境的实现方法

👉 首先用 `\newtheorem` 命令定义一个定理类环境

`\newtheorem{定理环境名}{标题}[主计数器]`

# 定理类环境的实现

## □ 定理类环境的实现方法

☞ 首先用 `\newtheorem` 命令定义一个定理类环境

```
\newtheorem{定理环境名}{标题}[主计数器]
```

☞ 调用该环境

```
\begin{定理环境名}[附加标题]
...
\end{定理环境名}
```

# 定理类环境的实现

## □ 定理类环境的实现方法

☞ 首先用 `\newtheorem` 命令定义一个定理类环境

```
\newtheorem{定理环境名}{标题}[主计数器]
```

☞ 调用该环境

```
\begin{定理环境名}[附加标题]
...
\end{定理环境名}
```

→ 以上定义的定理类环境使用独立的计数器



# 定理类环境的实现

## ❑ 多个定理类环境使用同一个计数器

```
\newtheorem{新定理环境名}[旧定理环境名] {标题}
```

- 👉 新定义的定理环境与已定义的 旧定理环境  
共用一个计数器 （例 901.tex）

# 定理类环境的实现

## ❑ 多个定理类环境使用同一个计数器

```
\newtheorem{新定理环境名}[旧定理环境名] {标题}
```

- 👉 新定义的定理环境与已定义的 旧定理环境 共用一个计数器 (例 901.tex)

## ❑ 定理环境缺省格式

- 👉 标题与编号用黑体, 正文用 *italic* 斜体
- 👉 含中文时, 一般需指定字体

# 定理环境宏包 amsthm

❑  $\mathcal{A}\mathcal{M}\mathcal{S}$ - $\text{\LaTeX}$  提供了一个加强定理环境功能的宏包

```
\usepackage{amsthm}
```

👉 新增命令 `\newtheorem*`, 创建不带编号定理环境

👉 提供 3 中预定义的定理格式

`plain`: 标题与编号均采用黑体, 正文用斜体

`definition`: 标题与编号为黑体, 正文正常字体

`remark`: 标题与编号为斜体, 正文用正常字体

👉 选择格式: `\theoremstyle{格式}`

➔ 在该命令后调用 `\newtheorem` 命令创建的定理环境, 将采用指定的格式

# 定理环境宏包 amsthm

👉 自定义定理类环境格式: `\newtheoremstyle`

→ 具体用法参见该宏包手册 `amsthm.pdf`

→ 详细示例参见 `thmtest.tex`

→ 例 `901.tex`

# 定理环境宏包 amsthm

👉 自定义定理类环境格式: `\newtheoremstyle`

→ 具体用法参见该宏包手册 `amsthm.pdf`

→ 详细示例参见 `thmtest.tex`

→ 例 `901.tex`

👉 也可是使用 `ntheorem` 宏包

# 定理环境宏包 amsthm

👉 自定义定理类环境格式: `\newtheoremstyle`

→ 具体用法参见该宏包手册 `amsthm.pdf`

→ 详细示例参见 `thmtest.tex`

→ 例 `901.tex`

👉 也可是使用 `ntheorem` 宏包

👉 `\swapnumber`

调用该命令后, 创建的定理类环境中编号显示在标题前

# 定理环境宏包 amsthm

👉 证明环境: `proof`

```
\begin{proof}[标题]  
...  
\end{proof}
```

- 若使用了标题, 则输出该标题, 否则输出 *Proof*.
- 证明环境结束时自动输出证毕记号 → □

# 定理环境宏包 amsthm

👉 证明环境: `proof`

```
\begin{proof}[标题]  
...  
\end{proof}
```

→ 若使用了标题, 则输出该标题, 否则输出 *Proof*.

→ 证明环境结束时自动输出证毕记号 → □

👉 证明命令: `\proof`

👉 证毕记号: `\qed`



# 定理环境宏包 amsthm

👉 证明环境: `proof`

```
\begin{proof}[标题]  
...  
\end{proof}
```

→ 若使用了标题, 则输出该标题, 否则输出 *Proof*.

→ 证明环境结束时自动输出证毕记号 → □

👉 证明命令: `\proof`

👉 证毕记号: `\qed`

→ 修改证毕记号: `\qedsymbol`

① 数学排版的国际标准

② 数学模式中的参数

③ 定理定义的排版

④ 巧妙使用阵列环境

⑤ 多行公式左列问题


# 巧妙使用阵列环境

## □ 阵列环境: `array`

- 👉 `array` 环境只能在数学模式中使用!
- 👉 可以使用阵列环境排版一些复杂的公式

# array 环境举例


$$|x| = \begin{cases} x, & x \geq 0 \\ -x, & x < 0 \end{cases}$$

 使用 array 环境(不自动编号)

```
|x|=\left\{
\begin{array}{rl}
x, & x\geq 0\\[2pt]
-x, & x<0
\end{array}
\right.
```

# array 环境举例

$$|x| = \begin{cases} x, & x \geq 0 \\ -x, & x < 0 \end{cases}$$

 使用 `amsmath` 宏包: `cases` 环境(不自动编号)

```
|x|=\begin{cases}
x, & x \geq 0 \\
-x, & x < 0
\end{cases}
```

## array 环境举例

$$|x| = \begin{cases} x, & x \geq 0 \\ -x, & x < 0 \end{cases} \quad \begin{matrix} (1) \\ (2) \end{matrix}$$

 使用 `cases` 宏包: `numcases` 环境(自动编号)

```
\begin{numcases} { |x| = }
      x, & $ x \ge 0 $ \\
     -x, & $ x < 0 $ \\
\end{numcases}
```



## 三角形矩阵

$$\begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ & a_{22} & \cdots & a_{2n} \\ & & \ddots & \vdots \\ 0 & & & a_{nn} \end{pmatrix}$$

---

```
\left(\begin{array}{cccc}
a_{11} & a_{12} & \cdots & a_{1n} \\
& a_{22} & \cdots & a_{2n} \\
& & \ddots & \vdots \\
0 & & & a_{nn}
\end{array}\right)
```

# array 环境举例

## 分块和加边矩阵

$$\left( \begin{array}{ccc|ccc} a & \cdots & a & b & \cdots & b \\ & \ddots & \vdots & \vdots & \ddots & \\ & & a & b & & \\ \hline & & 0 & c & \cdots & c \\ & & & \vdots & & \vdots \\ & & & c & \cdots & c \end{array} \right) \left. \begin{array}{l} \\ \\ \\ \\ \\ \end{array} \right\} \begin{array}{l} p \\ \\ \\ q \\ \\ \end{array}$$

$\underbrace{\hspace{10em}}_m \quad \underbrace{\hspace{10em}}_n$

(例 901.tex)



# array 环境举例

❑ 多行下标问题 
$$A = \sum_{\substack{1 \leq i \leq m \\ 1 \leq j \leq n \\ 1 \leq k \leq p}} a_{ijk}$$

→ 使用 array 环境

```
A=\sum_{\begin{array}{c} 1 \leq i \leq m \\ 1 \leq j \leq n \\ 1 \leq k \leq p \end{array}} a_{ijk}
```

# array 环境举例

❑ 多行下标问题

$$A = \sum_{\substack{1 \leq i \leq m \\ 1 \leq j \leq n \\ 1 \leq k \leq p}} a_{ijk}$$


---

→ 使用堆积命令: `\atop`

```
A=\sum_{
  {1\le i\le m}\atop
  {1\le j\le n \atop 1\le k\le p}
} a_{ijk}
```

# array 环境举例

👉 **amsmath** 宏包: `\substack` 命令或 `subarray` 环境

```
A=\sum_{\substack{
1 \leq i \leq m \\
1 \leq j \leq n \\
1 \leq k \leq p}} a_{ijk}
```

或

```
A=\sum\limits_{\scriptstyle
\begin{subarray}{l}
1 \leq i \leq m \\
1 \leq j \leq n \\
1 \leq k \leq p
\end{subarray}} a_{ijk}
```

## ① 数学排版的国际标准

## ② 数学模式中的参数

## ③ 定理定义的排版

## ④ 巧妙使用阵列环境

## ⑤ 多行公式左列问题

# 多行公式左列问题

## ❑ `\lefteqn{公式}`

- 👉 通常出现在 `eqnarray` 环境中  
用于解决左列太长的问题
- 👉 输出 `公式`, 但将其的长度视为零  
(类似于创建一个宽度为零的盒子)
- 👉 例 901.tex

# 第十一章

## 文档布局及相互联系

华东师范大学 数学系

2007 年 5 月



# 分栏

## 分栏

```
\documentclass[twocolumn]{article}
```

☞ 缺省为 `onecolumn`

☞ 选用 `twocolumn` 时, 可指定

- 栏间距: `\columnsep`  $\leftarrow$  `\setlength`
- 栏间分隔线粗细: `\columnseprule`, 缺省为 0

☞ 栏宽参数: `columnwidth`

- 通常不用设置, 但可以使用

♣ 若对整篇文档作修改, 可将设置命令放在导言区

# 分栏

## 部分文档分栏

```
\twocolumn[通栏文本]
```

👉 `\twocolumn`

→ 另起新页，以两栏格式排版其后面的文本

👉 `\onecolumn` → 新起一页，返回单栏格式

♣ 双栏排版时，先排满左栏，再排右栏

👉 例 1102.tex



# 分栏

## ❑ flushend 和 cuted 宏包

🔗 **flushend** 宏包: 使得选用 **twocolumn** 版式时, 双栏的**最后一页**上的左右两栏基本等高

- **\raggedend** → 取消平衡最后一页双栏的功能
- **\flushend** → 恢复平衡最后一页双栏的功能

🔗 **cuted** 宏包 → 在双栏版式中插入单栏文本

- 将单栏文本放入 **strip** 环境中
  - **cuted** 宏包与 **\onecolumn** 的区别: 不另起新页
- ♣ 此时使用 **\onecolumn** 和 **\twocolumn** 命令, 仍另起新页

# 分栏

## ❑ 多栏排版宏包: `multicol`

```
\begin{multicols}{栏数}
```

多栏文本

```
\end{multicols}
```

详细资料参见 `multicol.pdf`

●例 `1102.tex`

# 单双面

## ❑ 单双面选项: `oneside`, `twoside`

☞ 单面版式: 所有页码打印位置一致

双面版式: 页码打印在外侧 → 奇右偶左

☞ 单面版式: 左边距相同, 右边距相同

双面版式: 内侧边距相同, 外侧边距相同

☞ `article` 缺省是 `oneside`

`book` 缺省为 `twoside`

☞ 单面版式: 所有页面均称为右页

双面版式: 奇数页为右页, 偶数页为左页

# 单双面

## ❑ 单双面选项: `oneside`, `twoside`

☞ 双面版式:

每页正文底线位置在相同位置上 (`flushbottom`)

☞ 单面版式:

每页正文底线位置可稍有不同 (`raggedbottom`)

☞ `book` 类文档选项: `openany`, `openright`

控制每章的开始位置, 缺省为 `openright`.

# 与公式有关的选项

## □ 与公式有关的选项

👉 **leqno**: 公式编号位于左边

👉 **fleqn, mathindent**: 行间公式左对齐

# 页面布局

## □ 页面上排放文本区域的四个部分

- ☞ 页眉 (header)
- ☞ 页芯 (body)
- ☞ 页脚 (footer)
- ☞ 边注 (margin notes)

## □ 页面布局图 (见教材)

- ☞ `\textwidth`, `\textheight`
- ☞ `\hoffset`, `\voffset`
- ☞ `\oddsidemargin`, `\evensidemargin`

# 控制页面版式

## 控制页面版式的命令

```
\pagestyle{版式}
```

## 常用版式

- 👉 **plain** → 缺省选项, 页眉为空, 页码居中
- 👉 **empty** → 页眉页脚都为空, 不显示页码
- 👉 **headings** → 显示页眉 (含页码, 页眉标题)
  - 页眉标题由章, 节, 或小节标题组成, 自动提取
  - **book** 每一章的第一页和 **article** 的首页不显示页眉, 但在底部显示页码 (例 1103h1.tex, 1103hbook.tex)

# 控制页面版式

- **headings** 版式中页眉标题的组成

文档类		左页	右页
book, report	单面格式	——	章
	双面格式	章	节
article	单面格式	——	节
	双面格式	节	小节

👉 **myheadings** → 与 **headings** 类似, 但可以使用 `\markright` 或 `\markboth` 来定义页眉标题

👉 例 1103h2.tex



# 定义页眉标题

```
\markright{右页页眉标题}
```

```
\markboth{左页页眉标题}{右页页眉标题}
```

🔊 单页版式中使用 `\markright`

🔊 双页版式中使用 `\markboth`

🔊 缺省使用章节标题作为页眉标题

🔊 `headings` 版式中也可以使用这两个命令

🔊 `\leftmark` → 存放左页页眉标题

`\rightmark` → 存放右页页眉标题

♣ 改变当前页的页面版式: `\thispagestyle{版式}`

# 页码

## 指定页码样式

```
\pagenumbering{数字样式}
```

- ☞ arabic, roman, Roman, alph, Alph,
- ☞ 页码计数器: page
- ☞ 每调用一次 `\pagenumbering`, 计数器 page 置零
- ☞ book 类中三个控制版式命令:
  - `\frontmatter` → 通常放在前言和目录前
  - `\mainmatter` → 通常放在正文主体前
  - `\backmatter` → 通常放在参考文献和索引前

# 自定义页眉与页脚

## ❑ 自定义页眉与页脚的宏包: `fancyhdr`

```
\usepackage{fancyhdr}  
\pagestyle{fancy}
```

- 👉 该宏包将页面分为奇数页和偶数页
- 👉 页眉和页脚都被分为三个区域: 左区, 中区, 右区
  - 左区靠左边界, 中区居中, 右区靠右边界
- ♣ 各区内容不能太长, 否则会出现文本重叠

# 自定义页眉与页脚

## □ 自定义页眉与页脚的命令:

```
\fancyhead[选项]{内容}
```

```
\fancyfoot[选项]{内容}
```

👉 **选项** 为字母 L, C, R (代表区域) 和 O, E (代表奇偶页) 的组合

- 若 **选项** 中不含 **区域字母**, 则表示所有区域  
若 **选项** 中不含 **奇偶页字母**, 则表示所有页面

👉 简化的页眉页脚命令:

- \lhead, \chead, \rhead
- \lfoot, \cfoot, \rfoot



# 自定义页眉与页脚

## □ 首页显示页眉

在 `\maketitle` 后使用 `\thispagestyle{fancy}`

## □ 重新定义 `plain` 样式

- ☞ 一些 L<sup>A</sup>T<sub>E</sub>X 命令, 如 `\chapter`, `\maketitle` 等, 使用 `\thispagestyle` 来切换到 `plain` 样式
- ☞ `\fancypagestyle` 可重新定义现有页面样式或创建新的页面样式

`\fancypagestyle{页面样式名称}{页面格式}`

# 自定义页眉与页脚

## ❑ 使用章节标题作页眉

👉 重新定义命令 `\chaptermark`, `\sectionmark` 和 `\subsectionmark`

👉 `\leftmark` 和 `\rightmark`

## ❑ 用一行显示章的标题

例 1103hfancybook.tex

# 目录表

## 自动生成目录

\tableofcontents

## 👉 设置出现在目录中的章节层次

`\setcounter{tocdepth}{数}`

- 数 → 代表章节层次
- 为了生产正确的目录,通常需编译多次
- 只有自动编号的章节才会在目录中显示  
(即,带星号的章节环境不在目录中输出)



# 目录表

☞ 在目录中输出不编号的章节

```
\addcontentsline{toc}{章节名称}{条目内容}  
\addtocontents{toc}{条目内容}
```

- 章节名称 → `part`, `chapter`, `section`, ...

☞ 不同的章节层次采用不同的格式

☞ 第一个命令按章节名称的格式输出条目内容

第二个命令按原样输出条目内容, 无页码

# 插图目录与表格目录

## 插图目录与表格目录

```
\listoffigures, \listoftables
```

- 输出浮动环境 `figure` 和 `table` 生成的图表目录
- 图表目录中显示的为 `\caption` 中的内容
- 手工插入其它图表目录

```
\addcontentsline{清单类型}{格式}{条目内容}  
\addtocontents{清单类型}{格式}
```

清单类型: `toc` (目录), `lof` (插图), `lot` (表格)

格式: 章节名称, `figure`, `table`

# 目录清单标题中文化

## ❑ 目录清单标题中文化

```
\renewcommand{\contentsname}{\hei目\quad录}  
\renewcommand{\listfigurename}{\hei插图目  
录}  
\renewcommand{\listtablename}{\hei表格目  
录}
```

# 文档的分割处理

❑ 文档的分割处理: 将大文件分成几个小文件

❑ `\input` 命令: `\input{文件名}`

👉 将指定的文件读入到当前位置

👉 文件名可不带扩展名, 缺省为 `.tex`

👉 `\input` 命令可以出现在文档的任何部分

👉 `\input` 命令可以嵌套

👉 `\listfiles` → 放在导言区, 在屏幕和 log 文件中列出文件清单

👉 编译时, 读入所有文件, 再进行编译

# 文档的分割处理

❏ `\include` 命令: `\include{文件名}`

👉 相当于 `\clearpage\input{文件名}\clearpage`

👉 只能出现在正文部分

👉 `\include` 命令不可以嵌套

👉 优点: 与 `\includeonly{文件清单}` 配合使用

- 编译时, 只读入 `\includeonly` 指定的文件清单
- 保持有关页面, 章节, 公式编号等相关信息

♣ `\includeonly` 只能放在导言区

♣ 若没有 `\includeonly` 命令, 则读入所有文件

# 交叉引用

❑ 交叉引用: L<sup>A</sup>T<sub>E</sub>X 的一个非常有用的功能

👉 章节, 表格, 插图, 公式, 页码, 参考文献等

👉 L<sup>A</sup>T<sub>E</sub>X 自动跟踪所有的变动,

保证所有交叉引用与索引记录的正确性

❑ `\label{标记}`: 设置标记

👉 `\label` 命令常出现在章节, 公式, 定理类环境, 枚举罗列, `table`, `figure` 中的 `\caption` 参数中.

❑ `\ref{标记}`: 显示标记所对应的对象的编号

❑ `\pageref{标记}`: 显示标记所在页面的页码

# 索引

## 索引: 在导言区加入

```
\usepackage{makeidx}  
  
\makeindex
```

☞ `\index{索引条目}` → 标记索引条目

☞ `\printindex` → 输出索引条目

☞ 索引标题: `\indexname`

☞ 索引按ASCII码排序 (中文索引排序问题见教材)

☞ 多索引问题: `index` 宏包

# 第十二章

## 图形包介绍

华东师范大学 数学系

2007 年 5 月





# 图形包介绍

## □ L<sup>A</sup>T<sub>E</sub>X 常用图形包介绍

- ☞ 基本图形包: graphics, [graphicx](#)
  - 这两个宏包的命令和不可省参数均相同, 但可选参数的格式与内容不同
- ☞ 画交换图: amscd, Diagrams, [Xy-pic](#)
- ☞ 复杂图形: MetaPost, PSTricks, pgf
  - 这些图形包都采用坐标定位

# graphicx 宏包

## □ graphicx 宏包

```
\usepackage[选项]{graphicx}
```

选项参见教材

- 👉 **graphicx** 宏包主要用于插入外部图形 (彩色黑白)
- 👉 可以对插入的图形进行裁剪, 缩放, 翻转, 旋转
- 👉 L<sup>A</sup>T<sub>E</sub>X 支持最好的图形格式是 EPS (Encapsulated PostScript)

# EPS 图形文件

## ■ EPS 图形文件介绍

- 👉 PostScript 语言能够用来描述图形和文本
- 👉 传统的 PostScript(PS) 文件描述多页的文档
- 👉 Encapsulated PostScript(EPS) 文件描述插入文档的图形
- 👉 PS 和 EPS 主要的区别:
  - EPS 必须有 **BoundingBox** 来确定图形的大小
- 👉 指定一个 EPS 图形的 **BoundingBox** → **gsview**
- 👉 将其它图形文件转化成 EPS 文件  
→ **photoshop** 或 **imagemagick**

# graphicx 宏包

## ❑ 插入图形文件的基本命令

```
\includegraphics[选项]{图形文件名}
```

👉 图形文件名 中可以含路径

👉 常用选项

- **width** → 指定图形的宽度

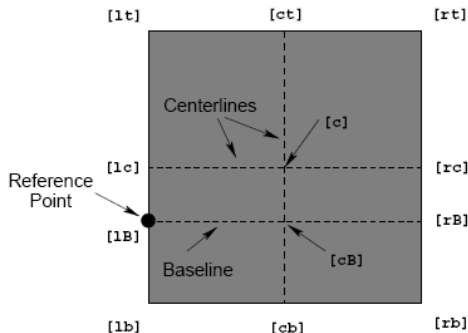
- height** → 指定图形的高度

- totalheight** → 指定图形的总高度

♣ 若只指定宽度或高度, 则按比例缩放图形

# graphicx 宏包

- **scale** → 缩放因子
- **angle** → 指定旋转角度, 逆时针, 以度为单位
- **origin** → 指定图形绕哪一点旋转, 缺省是图形的参考点



# graphicx 宏包

- **bb** → 设定 BoundingBox 的值, 例:  
bb=10 20 100 200 → 左下角 (10,20), 右上角 (100,200)
- **viewport** → 指定图形可看到的部分, 格式同 **bb**
- **trim** → 指定图形可以被看到的部分的另一选项, 所给四个数分别代表从左, 下, 右, 上被截去的值
- ♣ 以上两个选项通常与 **clip** 同时使用
- **clip** → 不显示图形在视图之外的部分
- **keepaspectratio** → 保持图形原有的宽高比例

# graphicx 宏包

## □ \includegraphics 使用举例

```
\includegraphics{tiger.eps}
```

```
\includegraphics[width=3cm]{tiger.eps}
```

```
\includegraphics[width=0.8\textwidth]{tiger.eps}
```

```
\includegraphics[width=\textwidth-4cm]{tiger.eps}1
```

---

<sup>1</sup>需加载 `calc` 宏包





## \includegraphics 使用举例



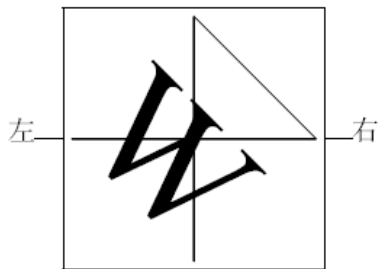
例三

```
左\HR\fbox{%
  \includegraphics
    [height=20mm,width=30mm]%
    {w.eps}}\HR 右
```

例四

```
左\HR\fbox{%
  \includegraphics
    [height=20mm,width=30mm,%
    keepaspectratio]{w.eps}}%
\HR 右
```

# \includegraphics 使用举例



例五

```

左\HR\fbbox{%
  \includegraphics
    [angle=-45]{w.eps}}%
\HR 右

```



# 旋转和缩放对象

## ❑ 旋转和缩放任意 L<sup>A</sup>T<sub>E</sub>X 对象：文本, EPS 等

```
\scalebox{横向缩放因子}[竖向缩放因子]{对象}  
\resizebox{宽度}{高度}{对象}  
\resizebox*{宽度}{总高度}{对象}  
\rotatebox[选项]{角度}{对象}
```

👉 旋转和缩放插图时, 通常只需使用  
`\includegraphics`

# 旋转和缩放对象

`\scalebox{横向缩放因子}[竖向缩放因子]{对象}`

- ☞ 若省略 竖向缩放因子, 则表示横向与竖向使用相同的缩放因子
- ☞ 若缩放因子是负数, 则表示对对象进行反射
- ♣ 转化成 ps 或 pdf 文件后才能看出效果

# 旋转和缩放对象

```
\resizebox{宽度}{高度}{对象}
```

```
\resizebox*{宽度}{总高度}{对象}
```

☞ 将对象缩放到指定的大小

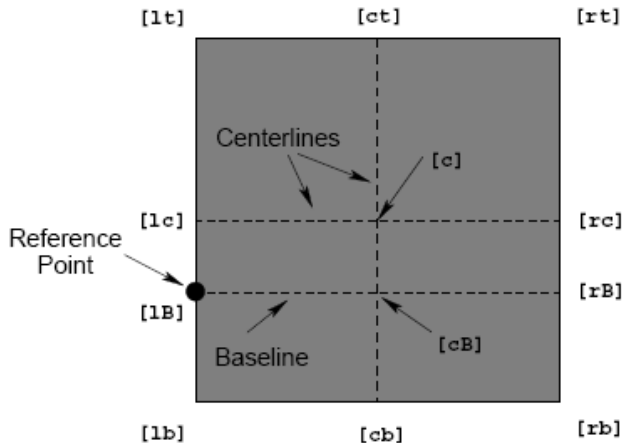
☞ 若宽度和高度中有一个是 !, 则表示保持对象的原有横竖比例

# 旋转和缩放对象

`\rotatebox[选项]{角度}{对象}`

- ☞ 将对象按逆时针旋转一给定度数的角度
- ☞ 可以在选项中指出对象绕给定的点来旋转  
缺省为对象的参考点
  - $x=x, y=y \rightarrow$  指定的点坐标为  $(x, y)$
  - `origin` 中的十二个点之一

# graphicx 宏包





# 第十三章

## 输出到投影仪(1)

华东师范大学 数学系

2007 年 5 月



# 本讲内容

## □ 引言

- ☞ xcolor 宏包
- ☞ pdf $\text{T}_\text{E}$ X 和 pdf $\text{L}_\text{A}$ T $\text{E}$ X
- ☞ hyperref 宏包

## □ 制作幻灯片

- ☞ slides 文档类
- ☞ pdfscreen 宏包
- ☞ beamer 宏包(下一讲)

# 利用 L<sup>A</sup>T<sub>E</sub>X 制作幻灯片

## □ 使用 L<sup>A</sup>T<sub>E</sub>X 制作幻灯片的宏包

- 👉 slides 文档类
- 👉 pdfscreen、beamer、texpower、prosper  
context、pdfslides、seminer、foitex ...
- 👉 More on screen presentation tools  
<http://www.miwie.org/presentations/presentations.html>

# xcolor 宏包

## □ xcolor 宏包: 显示或打印彩色

```
\usepackage[选项]{xcolor}
```

选项主要包括

- 彩色驱动程序名: dvips, dvipdfm, pdftex, ...
- 彩色模式: [natural](#), rgb, cmyk, ...<sup>1</sup>
- 预定义色彩集合: dvipsnames, dvipsnames\*, ...  
[red](#), [green](#), [blue](#), [yellow](#), [cyan](#), [magenta](#), [black](#),  
[white](#), [orange](#), [violet](#), [purple](#), [brown](#), [pink](#), ...

👉 例: dvipsnames 预定义的 68 种色彩(1301.tex)

---

<sup>1</sup>更多模式见下页

# 彩色模式

Table : Supported color models

<i>Name</i>	<i>Base colors/notions</i>	<i>Parameter range</i>	<i>Default</i>
<b>rgb</b>	<i>red, green, blue</i>	$[0, 1]^3$	
<b>cmY</b>	<i>cyan, magenta, yellow</i>	$[0, 1]^3$	
<b>cmyk</b>	<i>cyan, magenta, yellow, black</i>	$[0, 1]^4$	
<b>hsb</b>	<i>hue, saturation, brightness</i>	$[0, 1]^3$	
<b>Hsb</b>	<i>hue<sup>o</sup>, saturation, brightness</i>	$[0, H] \times [0, 1]^2$	$H = 360$
<b>tHsb</b>	<i>hue<sup>o</sup>, saturation, brightness</i>	$[0, H] \times [0, 1]^2$	$H = 360$
<b>gray</b>	<i>gray</i>	$[0, 1]$	
<b>RGB</b>	<i>Red, Green, Blue</i>	$\{0, 1, \dots, L\}^3$	$L = 255$
<b>HTML</b>	<i>RRGGBB</i>	$\{000000, \dots, \text{FFFFFF}\}$	
<b>HSB</b>	<i>Hue, Saturation, Brightness</i>	$\{0, 1, \dots, M\}^3$	$M = 240$
<b>Gray</b>	<i>Gray</i>	$\{0, 1, \dots, N\}$	$N = 15$
<b>wave</b>	<i>lambda (nm)</i>	$[363, 814]$	

$L, M, N$  are positive integers;  $H$  is a positive real number

# xcolor 宏包

## □ 使用彩色

`\color{色彩名}` → 声明

`\textcolor{色彩名}{文本}` → 命令

`\pagecolor{色彩名}` → 声明, 指定背景色<sup>2</sup>

👉 色彩名 必须是已经有定义的

👉 可以使用不同色彩的混合, 如:

`\color{green!40!yellow}`

→ 40% 的 green 与 60% 的 yellow 的混合

`\color{-green!40!yellow}` → 上述颜色的补色

<sup>2</sup>背景色需要转化成 ps 或 pdf 后才能看出效果

# xcolor 宏包

## ■ 色彩的混合公式

$$\begin{aligned} & C_0!P_1!C_1!P_2!\dots!P_n!C_n \\ &= p_n \cdots p_1 \cdot C_0 + p_n \cdots p_2(1 - p_1) \cdot C_1 \\ &\quad + p_n \cdots p_3(1 - p_2) \cdot C_2 + \cdots \\ &\quad + p_n(1 - p_{n-1}) \cdot C_{n-1} + (1 - p_n) \cdot C_n \end{aligned}$$

其中  $P_k \in [0, 100]$  且  $p_k = P_k/100, k = 1, \dots, n$

# xcolor 宏包

## ❑ 使用彩色的另一种用法

```
\color[色彩模式]{数据}  
\textcolor[色彩模式]{数据}{文本}  
\pagecolor[色彩模式]{数据}
```

👉 例:

- `\textcolor[rgb]{0.3,0.8,0.1}`{第一种新颜色}
- `\textcolor[cmyk]{0.0,0.5,0.1,0.5}`{第二种新颜色}



# xcolor 宏包

## □ 定义色彩

```
\definecolor{色彩名}{色彩模式}{数据}  
\providecolor{色彩名}{色彩模式}{数据}
```

- 常用的色彩模式有: rgb, cmyk, gray 等
- 数据: 一组用逗号分隔的十进制小数, 表示每个分量的力度, 如:  

```
\definecolor{red}{rgb}{1,0,0},  
\definecolor{yellow}{cmyk}{0,0,1,0}
```
- 如果色彩名已有定义, `\providecolor` 不起作用, 而 `\definedcolor` 则覆盖原来的定义

# xcolor 宏包

## ■ 彩色盒子

```
\colorbox{色彩名}{文本}
```

```
\fcolorbox{色彩一}{色彩二}{文本}
```

```
\colorbox[色彩模式]{数据}{文本}
```

```
\fcolorbox[色彩模式]{数据一}{数据二}{文本}
```

## ■ \normalcolor: 恢复正常色

# 如何使用 pdfTeX

- `dvipdfm` → 直接将 dvi 转化成 pdf  
<http://gaspra.kettering.edu/dvipdfm>
- `dvipdfmx` → `dvipdfm` 的一个扩展  
<http://project.ktug.or.kr/dvipdfmx>
- `pdfTeX` → 处理 Plain TeX 和  $\mathcal{A}\mathcal{M}\mathcal{S}$ -TeX 文件
- `pdfLATeX` → 处理 L<sup>A</sup>TeX 文件
- ♣ `pdfTeX` 和 `pdfLATeX` 直接生成 pdf 文件, 可以增加一些 L<sup>A</sup>TeX 中没有的 pdf 功能
- ♣ 支持的图片: pdf, png, jpg, tif, 但不支持 eps

# 书签

## ❑ 建立书签: hyperref 宏包

```
\usepackage[选项]{hyperref}
```

```
\usepackage[选项]{hyperref}
```

```
\hypersetup{ ... }
```

例:

```
\usepackage[dvipdfmx,CJKbookmarks]{hyperref}
```

```
\usepackage[dvipdfmx]{hyperref}
```

```
\hypersetup{CJKbookmarks=true}
```

# 书签

## □ hyperref 宏包的常用选项:

- ☞ 驱动选项: dvipdfmx, dvips, pdftex, ...<sup>3</sup>
- ☞ pdf 属性: pdftitle, pdfauthor, pdfsubject, ...
- ☞ pdf 打开方式: pdfstartview, pdfpagemode, ...
- ☞ 书签: bookmarks, CJKbookmarks, bookmarksnumbered, bookmarksopen, ...
- ☞ 链接属性: colorlinks, linkcolor, citecolor, ...
- ☞ 更多选项参见 [hyperref manual](#)
- ☞ 例: 1302.tex

---

<sup>3</sup>放在 `\hypersetup` 中不起作用

# 制作幻灯片 — slides 文档类

## □ slides 文档类: 制作透明胶片

```
\documentclass[选项]{slides}
```

- ☞ 字体较大, `\normalsize` → 20pt
- ☞ 缺省字体为 `sffamily`
- ☞ 三种环境: `slide`, `overlay`, `note`
- ☞ 一个简单的例子: 1303slides.tex

# 制作幻灯片 — pdfscreen

## □ pdfscreen 宏包

```
\usepackage[选项]{pdfscreen}
```

### 👉 常用选项

- screen: 产生适合屏幕输出的 pdf 文件
- print: 输出打印版
- panelleft, panelright, nopanel: 设置导航面板
- paneltoc: 将目录放在导航面板上
- sectionbreak: 每节分页
- 导航面板配色方案: blue, gray, orange, ...

♣ pdfscreen 宏包自动输入 graphicx 和 color 宏包

# 制作幻灯片 — pdfscreen

## □ pdfscreen 定制幻灯片命令(出现在导言区)

- ☞ `\screensize{高度}{宽度}`: 屏幕大小(必须)
- ☞ `\margins{左}{右}{上}{下}`: 定义页边空白<sup>4</sup>
- ☞ `\overlay{pdf 图形文件}`: 设置幻灯片背景
- ☞ `\paneloverlay{pdf 图形文件}`: 导航面板背景
- ☞ `\urlid{URL 地址}`: 设置主页链接地址
- ☞ `\emblema{图形文件}`: 在导航面板加入图片
- ☞ 导航按钮: `\bottombuttons`, `\nobottombuttons`, `\topbuttons`, `\notopbuttons`

---

<sup>4</sup>`\margins` 最好出现在 `\screensize` 的前面



# 制作幻灯片 — pdfscreen

## □ pdfscreen 导航面板按钮中文化

- ☞ \panelhomepagename: 主页
- ☞ \paneltitlepagename: 标题页
- ☞ \panlabstractname: 摘要页
- ☞ \panelcontentsname: 目录页
- ☞ \panelgobackname: 返回
- ☞ \panelfullscreenname: 全屏显示
- ☞ \panelclosename: 关闭
- ☞ \panelquitname: 退出
- ☞ \panelpagename, \panelofname, \panelofnamep  
第?页共?页

# 制作幻灯片 — pdfscreen


## □ slide 环境

```
\begin{slide}  
  
...  
  
\end{slide}
```

👉 一个 slide 环境生成一张幻灯片

# 制作幻灯片 — pdfscreen

## ❑ 页面过渡命令<sup>5</sup>

 `\pageTransitionBlindsH,`  
`\pageTransitionBlindsV,`  
`\pageTransitionSplitVI,`  
`\pageTransitionDissolve,`  
`\pageTransitionSplitVO,`  
`\pageTransitionSplitHI,`  
`\pageTransitionSplitHO,`  
`\pageTransitionGlitter{270},`  
`...`

---

<sup>5</sup>需 `texpower` 宏包: `\usepackage[display]{texpower}` 

# 第十三章

## 输出到投影仪(2)

华东师范大学 数学系

2007 年 6 月



# 利用 L<sup>A</sup>T<sub>E</sub>X 制作幻灯片

- 使用 L<sup>A</sup>T<sub>E</sub>X 制作幻灯片的宏包 — beamer
- beamer 是目前制作幻灯片的最佳宏包之一
- 最全的学习资料 — beamer 手册

# 利用 L<sup>A</sup>T<sub>E</sub>X 制作幻灯片

- ❑ 使用 L<sup>A</sup>T<sub>E</sub>X 制作幻灯片的宏包 — beamer
- ❑ beamer 是目前制作幻灯片的最佳宏包之一
- ❑ 最全的学习资料 — beamer 手册

# 利用 L<sup>A</sup>T<sub>E</sub>X 制作幻灯片

- ❑ 使用 L<sup>A</sup>T<sub>E</sub>X 制作幻灯片的宏包 — beamer
- ❑ beamer 是目前制作幻灯片的最佳宏包之一
- ❑ 最全的学习资料 — beamer 手册

## □ Beamer 是 L<sup>A</sup>T<sub>E</sub>X 制作 Slide 的一个文档类

- ☞ 用 pdf<sub>l</sub>atex 编译, 不需要其它后处理程序
- ☞ 同标准 L<sup>A</sup>T<sub>E</sub>X 结合度高
- ☞ 直观而方便的创建遮挡 (overlays) 及其它各种漂亮的演示效果
- ☞ 利用 theme 实现内容和风格的独立 可以轻松地改变风格
- ☞ 例: 1304beamer01.tex



## □ 在 L<sup>A</sup>T<sub>E</sub>X 中使用 beamer 的一般步骤

- 👉 将 L<sup>A</sup>T<sub>E</sub>X 的文档类型从 article 改为 beamer
- 👉 用 section 和 subsection 组织 L<sup>A</sup>T<sub>E</sub>X 的文档结构
- 👉 用 frame 环境制作独立的帧
- 👉 执行 pdf<sub>l</sub>atex 编译 tex 文件

# beamer

## beamer 文档类

```
\documentclass[11pt,CJK,compress]{beamer}  
\usetheme{Warsaw}  
\usecolortheme{lily}  
... ..
```

## 幻灯片主体结构

```
\section{ ... }      % 用于书签和导航条  
\subsection{ ... }  % 用于书签和导航条  
\begin{frame}  
... ..  
\end{frame}
```

# beamer

## ❑ beamer 中的帧 (frame)

```
\begin{frame}[选项]
  \frametitle{...} → 指定帧的标题
  \subframetitle{...} → 指定帧的副标题
  ...
\end{frame}
```

- 👉 帧通常由若干幅幻灯片组成
- 👉 一个帧中可以包含覆盖, 逐段显示等内容
- 👉 一个有用的选项: `plain`
- 👉 将所有内容放入到 `frame` 环境中

# beamer

## ■ 标题页

```
\begin{frame}  
  \titlepage → 生成标题页  
\end{frame}
```

👉 标题页通常包含: 标题, 副标题, 报告人, ...

- `\title{...}`
- `\subtitle{...}`
- `\author{...}`
- `\date{...}`
- `\institute{...}`
- `\titlegraphic{...}`

# beamer

## ■ 目录页

```
\begin{frame}  
  \frametitle{Outline}  
  \tableofcontents[选项] → 生成目录  
\end{frame}
```

☞ `\tableofcontents` 中的可选项

- `pausesections` → 逐段显示目录
- `hideallsubsections`, `hideothersubsections`
- `currentsection`, `currentsubsection`

☞ 若不想使用自动生成的目录, 可以手工输入

# 中文支持 – CJK 宏包

## 中文模板

```
\documentclass [CJK] {beamer}
... ..
\usepackage{CJK}
... ..
\begin{document}
\begin{CJK}{GBK}{song}
... ..
\end{CJK}
\end{document}
```

# beamer

## □ beamer 中的一些预定义环境

👉 theorem, lemma, corollary, definition, ...

### Theorem (费马)

不存在整数  $x, y, z$  以及大于 2 的整数  $n$ , 使得  $x^n + y^n = z^n$  成立.

👉 block 环境 → 类似定理环境, 可自选标题

### block 环境

```
\begin{block}{标题}  
    ...  
\end{block}
```

# beamer

## 🔑 columns 环境

```
\begin{columns}[选项]  
  \column{宽度}  
  ... ..  
  \column{宽度}  
  ... ..  
\end{columns}
```



## ■ beamer 中的罗列环境

- 👉 幻灯片中应多使用罗列环境
- 👉 可以嵌套使用, 但通常不要超过 2 层
- 👉 罗列条目的自动逐段显示

`\beamerdefaultoverlayspecification{<+>}`

- 👉 手工指定逐段显示的内容: <数字范围>  
如: <3>, <-3>, <2-4>, <2->

# beamer

## ❑ beamer 中的抄录环境: `semiverbatim`

```
\begin{semiverbatim}  
    . . .  
\end{semiverbatim}
```

- 👉 使用该环境时需在 `frame` 后加选项 `[fragile]`  
或 `[containsverbatim]`
- 👉 `\`, `{`, `}` 仍然保持特殊含义
- 👉 `verbatim` 仍然可以使用

# 参考文献: thebibliography

🔗 条目中不同性质内容用 `\newblock` 分隔

## bibitem 条目示例

```
\bibitem{Golub96}
G.H. Golub and C. Van loan,
\newblock {\em Matrix Computations, 3rd},
\newblock
The Jhon Hopkins University press, 1996
```

📄 G.H. Golub and C. Van loan,  
*Matrix Computations, 3rd*,  
 The Jhon Hopkins University press, 1996

# 插入图片: \includegraphics

## 插入图片

```
\includegraphics[scale=0.5]{tiger_small.png}
```



- 可插入 pdf, png, jpg 等图片, 但不支持 eps 图片

# 插入多媒体: multimedia 宏包

```
\usepackage{multimedia}
```

## 插入电影

```
\movie[autostart]{}{mathproof.wmv}
```

## 插入声音

```
\sound[autostart]{}{Take_me_to_you_heart.wav}
```

- 所能播放的声音文件类型要看 adobe reader 是否支持
- 这里是用 `\movie` 播放的音乐

# 插入 logo

## 插入 logo

```
\pgfdeclareimage[height=.8cm]{logo}{math}  
\logo{\pgfuseimage{logo}}
```

- 在幻灯片的右下角插入 logo

# 幻灯片切换

## 幻灯片切换

```
\transblindshorizontal  
\transblindsvertical  
\transboxin  
\transboxout  
\transdissolve  
\transglitter  
\transsplithorizontalin  
\transsplitverticalin  
\transsplithorizontalout  
\transsplitverticalout  
\transwipe
```

# 导航图标



✎ `\setbeamertemplate{navigation symbols}%  
{\insert...navigation symbol}`

- 其中... 可以为:

slide, frame, subsection, section, doc, backfindforward

## Example

```
\setbeamertemplate{navigation symbols}%
{\insertdocnavigation symbol}
```

- 指定多个导航图标时, 用逗号分隔

✎ 取消导航图标

```
\setbeamertemplate{navigation symbols}{}.
```



# 导航图标



✎ `\setbeamertemplate{navigation symbols}%  
{\insert...navigationsymbol}`

- 其中 ... 可以为:

slide, frame, subsection, section, doc, backfindforward

## Example

```
\setbeamertemplate{navigation symbols}%
{\insertdocnavigationsymbol}
```

- 指定多个导航图标时, 用逗号分隔

✎ 取消导航图标

```
\setbeamertemplate{navigation symbols}{}%
```

# 导航图标



✚ `\setbeamertemplate{navigation symbols}%  
{\insert...navigationsymbol}`

- 其中 ... 可以为:

slide, frame, subsection, section, doc, backfindforward

## Example

```
\setbeamertemplate{navigation symbols}%
{\insertdocnavigationsymbol}
```

- 指定多个导航图标时, 用逗号分隔

✚ 取消导航图标

```
\setbeamertemplate{navigation symbols}{{}}
```

# beamer

## 覆盖的技巧

👉 在命令后加覆盖指示, 如:

```
\textbf<2-3>{\cdots}, \color<2->{blue}{\cdots}
```

👉 在环境后加覆盖指示, 如:

```
\begin{theorem}<2->
  \cdots \cdots
\end{theorem}
```

👉 \uncover, \only, \visible, \onslide

👉 \setbeamercovered{transparent}

→ 使覆盖部分变得透明

## □ 覆盖的技巧

☞ `\pause`

☞ `\uncover` 控制内容什么时候显示

☞ `\only` 内容不显示时完全消失(不占版面)

☞ `\visible` 同 `\uncover`, 但 `\setbeamercovered` 不起作用

☞ `\onslide` 其作用范围不受  $\text{\TeX}$  群组的约束

## □ beamer 的五类主题样式

- 演示主题: 幻灯片模板
- 色彩主题
- 字体主题
- 内部主题
- 外部主题

## ■ 演示主题 (presentation)

👉 `\usetheme`[option]{主题名}

👉 旧主题: bars, boxes, classic, default, lined, plain, shadow, sidebar, sidebardark, ...

👉 新主题 (v 3.0)

- 没有导航条: default, Pittsburgh, Rochester, ...
- 带树形导航条: Antibes, JuanLesPins, Montpellier
- 带目录的侧边导航条: Berkeley, PaloAlto, ...
- 带微型导航条: Berlin, Ilmenau, Singapore, ...
- 带节小节标题: Copenhagen, Luebeck, Warsaw, ...


# beamer

## □ 色彩主题 (color)

 `\usecolortheme[option]{主题名}`

- default, structure, sidebartab, albatross, beetle, crane, seahorse, fly, seagull, lily, orchid, rose, whale, dolphin

## □ 字体主题 (font)

 `\usefonttheme[option]{主题名}`

- default, professionalfonts, serif, structurebold, structureitalicserif, structuresmallcapsserif

# beamer

## 内部主题 (inner)

☞ title and part, itemize, enumerate, description, block environments, theorem and proof, figures and tables, footnotes, bibliography

☞ `\useinnertheme[option]{主题名}`

- default, circles, rectangles, rounded, inmargin

## 外部主题 (outer)

☞ head- and footline, sidebars, logo, frame title

☞ `\useoutertheme[option]{主题名}`

- default, professional fonts, serif, structurebold, structureitalicserif, structuresmallcapsserif



# 对 Frame 内容的建议

- ❗ 内容少了比多了好
- ❗ 不要假定每个听众都是专家。一些特定的名词要给出解释
- ❗ 不要把整个演讲过程中都不会讲到的东西放在 Slide 中
- ❗ 尽量保持简单

# 对标题的建议

- ☞ 每一幅都要有标题
- ☞ 标题要反映内容, 让人一看就知道内容是什么, 尽量用清楚的表达
- ☞ 如果可能, 前后的小标题要连贯, 这些标题本身就能反映事情
- ☞ 英语标题的大写问题, 或者都用大写(除了a,the等连词), 或者都用小写(除了第一个字母), 要前后统一
- ☞ 英语的全文标题要大写, 不管其它的内容是否大写

# 对正文的建议

- ✎ 永远不要为了在一幅中放更多的内容而选用小字体
- ✎ 尽量用列表或分类, 而少用大片无结构的文本, 不要用长句子
- ✎ 不要用连字符, 如果确实需要, 手工使用 - 命令
- ✎ 手工换行, 根据句子的逻辑换行
- ✎ 图表中的文本和数字应该同正文的同等大小, 难以分辨的坐标轴数字经常使图表变得无意义

# 对图表的建议

- ☞ 每页最少有一个图表, 可视的效果对听众有很大帮助
- ☞ 一般情况下把图放在文字的左边(使用column环境)
- ☞ 图的风格应该跟正文保持协调。包括字体大小, 最小点的大小, 线的宽度等
- ☞ 图表颜色要统一
- ☞ 同文本一样, 图表中出现的東西都要解释. 未解释的细节会使听众迷惑, 当使用论文或其它地方的图时一定要小心, 因为一般来说这些图可能含有一些你无法解释的细节

# 对色彩的建议

- 不要频繁的指定颜色
- 在白背景下使用亮的颜色要小心
- 尽量用大反差, 正文和背景的反差一定要大
- 渐变的背景会降低文字的易读性
- 反色效果(黑背景亮文本)在一个比较亮的环境下放映可能出问题, 并且打印不方便

# 对动画和特殊效果的建议

- 使用动画来演示动态系统或算法
- 不要仅仅为了吸引观众而使用动画, 这通常会分散听众对报告主题的关注
- 不要使用分散注意力的特殊效果, 除非你有足够的时间用它